REVISION HANDOUTS

AS-Level Handouts

Every AS topic across all eight subjects.

AS 。

Bilingual study handouts • 74 topics.

Contents

AS-Level Physics 5

1 Physical quantities and units 5

2 Kinematics 22

3 Dynamics 36

4 Forces, density and pressure 52

5 Work, energy and power 67

6 Deformation of solids 80

7 Waves 93

8 Superposition 108

9 Electricity 123

10 D.C. circuits 136

11 Particle physics 153

AS-Level Chemistry 169

1 Atomic structure 169

2 Atoms, molecules and stoichiometry 181

3 Chemical bonding 190

4 States of matter 198

5 Chemical energetics 205

6 Electrochemistry 211

7 Equilibria 216

8 Reaction kinetics 224

9 The Periodic Table: chemical periodicity 229

10 Group 2 235

11 Group 17 239

12 Nitrogen and sulfur 245

13 An introduction to AS Level organic chemistry 250

14 Hydrocarbons 260

15 Halogen compounds 267

16 Hydroxy compounds 272

17 Carbonyl compounds 278

18 Carboxylic acids and derivatives 282

19 Nitrogen compounds 286

20 Polymerisation 290

21 Organic synthesis 294

22 Analytical techniques 299

AS-Level Biology 304

1 Cell structure 304

2 Biological molecules 313
3 Enzymes 323
4 Cell membranes and transport 332
5 The mitotic cell cycle 340
6 Nucleic acids and protein synthesis 346
7 Transport in plants 354
8 Transport in mammals 361
9 Gas exchange 369
10 Infectious diseases 374
11 Immunity 379

AS-Level Computer Science 387

1 Information representation 387
2 Communication 403
3 Hardware 421
4 Processor Fundamentals 441
5 System Software 459
6 Security, privacy and data integrity 473
7 Ethics and Ownership 487
8 Databases 497
9 Algorithm Design and Problem-solving 513
10 Data Types and Structures 528
11 Programming 544
12 Software Development 568

AS-Level Economics 583

1 Basic economic ideas and resource allocation 583

2 The price system and the microeconomy 597
3 Government microeconomic intervention 613
4 The macroeconomy 626
5 Government macroeconomic intervention 646
6 International economic issues 658

AS-Level Business 673

1 Business and its environment 673
2 Human resource management 687
3 Marketing 696
4 Operations management 707
5 Finance and accounting 714

AS-Level Mathematics 725

1 Pure Mathematics 1 725
2 Pure Mathematics 2 745
4 Mechanics 753
5 Probability & Statistics 1 760

AS-Level Further Mathematics 768

1 Further Pure Mathematics 1 768
3 Further Mechanics 775
4 Further Probability & Statistics 783

Physical quantities and units

A-Level Physics

Physical quantities

A **physical quantity** has two parts: a number (its **magnitude**) and a **unit**. The number on its own tells you nothing. You must also say what is measured and in which unit.

Example: "the length is 1.5" is not complete. "The length is 1.5 m" is a physical quantity.

Making estimates

You should be able to **estimate** the size of the physical quantities in this syllabus. Paper 1 usually opens with a question like this. Learn these rough values:

- **mass** of an adult human: ~ 70 kg; of a car: ~ 1000 kg; of an apple: ~ 0.1 kg
- **weight** of an adult human: ~ 700 N; of an apple: ~ 1 N
- height of an adult human: ~ 1.7 m; height of a room: ~ 3 m
- walking speed: ~ 1.5 m s^{-1}; a car on a fast road: ~ 30 m s^{-1}
- speed of sound in air: ~ 340 m s^{-1}; speed of light in a **vacuum**: 3.0×10^8 m s^{-1}
- **acceleration** of **free fall**: $g \approx 9.81$ m s^{-2}
- **density** of water: 1000 kg m^{-3}; of air: ~ 1.2 kg m^{-3}; of steel: ~ 8000 kg m^{-3}
- **atmospheric pressure**: $\sim 1.0 \times 10^5$ Pa
- room temperature: ~ 20 °C ≈ 293 K
- power of a kettle: ~ 2 kW; of a person climbing stairs: ~ 300 W
- wavelength of visible light: $\sim 5 \times 10^{-7}$ m; diameter of an atom: $\sim 10^{-10}$ m; of a nucleus: $\sim 10^{-15}$ m

A good estimate has the right **order of magnitude** (the right power of ten). For a human, 70 kg is a good guess; 7 kg is not.

To estimate a quantity that is not in the list, build it from ones that are. The kinetic energy of a moving car is $\frac{1}{2}mv^2 \approx \frac{1}{2} \times 1000 \times 30^2 \approx 5 \times 10^5$ J. The pressure under a standing person is 700 N/0.02 m^2 $\approx 4 \times 10^4$ Pa. Always check that the power of ten looks sensible before you move on.

SI units

Base units

The SI system has seven **base quantities**; five of them are used in this topic. You must know these five and their units:

- **mass** – kilogram, kg
- **length** – metre, m
- time – second, s
- **current** – ampere , A
- **temperature** – kelvin , K

Every other unit in this syllabus is built from these five.

Derived units

A **derived unit** is made by multiplying or dividing base units. You should be able to write any quantity in this syllabus in base units.

Build a derived unit from the equation that defines it:

- **speed** = distance / time, so its unit is m s^{-1}
- **acceleration** = change in **velocity** / time, so its unit is m s^{-2}
- **force** = mass × acceleration, so its unit is kg m s^{-2}. The **newton** is 1 N = 1 kg m s^{-2}.
- **work** and **energy** = force × distance, so the unit is kg m^2 s^{-2}. The **joule** is 1 J = 1 kg m^2 s^{-2}.
- **power** = energy / time, so the unit is kg m^2 s^{-3}. The **watt** is 1 W = 1 kg m^2 s^{-3}.
- **pressure** and **stress** = force / area, so the unit is kg m^{-1} s^{-2}. The **pascal** is 1 Pa = 1 kg m^{-1} s^{-2}.

When a question asks for the SI base units of a quantity, replace each named unit with its base units, then simplify. Example: the SI base units of the watt are kg m^2 s^{-3}.

Checking that the units match

An equation is **homogeneous** when both sides have the same base units. In plain words: the units on both sides match.

Write each side in base units and compare. Take the equation $v^2 = u^2 + 2as$:

- left side: $(\text{m s}^{-1})^2 = \text{m}^2\ \text{s}^{-2}$
- right side, first term: $(\text{m s}^{-1})^2 = \text{m}^2\ \text{s}^{-2}$
- right side, second term: $\text{m s}^{-2} \cdot \text{m} = \text{m}^2\ \text{s}^{-2}$

Both sides give $\text{m}^2\ \text{s}^{-2}$, so the units match.

Be careful: matching units do **not** prove the whole equation is correct. It could still have a wrong number, or a missing factor of 2. But if the units do **not** match, the equation is wrong for sure.

> **Worked example.** The drag force on a falling ball is given by $F = kv^2$, where v is the speed. Find the SI base units of the constant k.

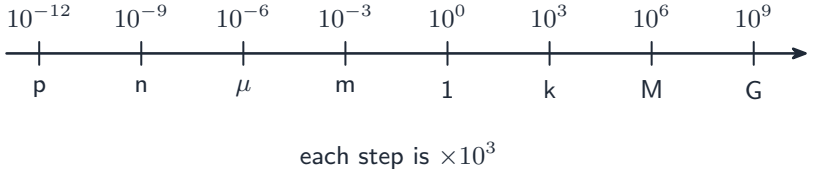
Rearrange: $k = F/v^2$. In base units, F is kg m s^{-2} and v^2 is m^2 s^{-2}, so

$$k: \ \frac{\text{kg m s}^{-2}}{\text{m}^2\ \text{s}^{-2}} = \text{kg m}^{-1}.$$

A multiple-choice question often asks "which equation could be correct?". Check the base units of each option; only a homogeneous equation can be correct. A pure number (like 2, π or $\frac{1}{2}$) has no unit, so it never changes the check.

Prefixes

A **prefix** is a letter put in front of a unit to make it bigger or smaller by powers of ten. You must know these:



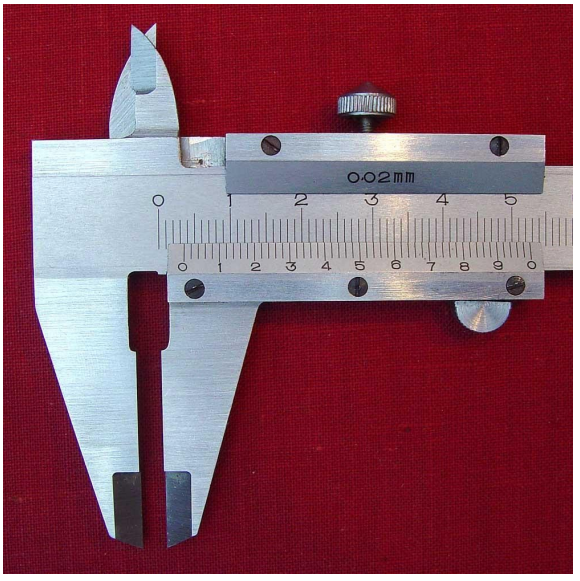
The SI prefixes climb in steps of a thousand, from pico to giga

Prefix	Symbol	Factor
tera	T	10^{12}
giga	G	10^{9}
mega	M	10^{6}
kilo	k	10^{3}
deci	d	10^{-1}
centi	c	10^{-2}
milli	m	10^{-3}
micro	μ	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}

To change a prefixed unit into base units, replace the prefix with its factor, then simplify. Example: change 0.25 kN mm^{-2} into N m^{-2}:

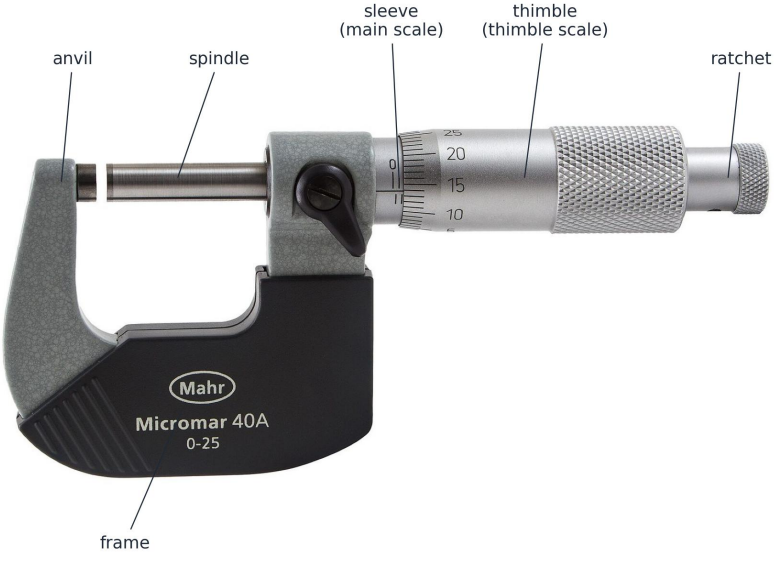
$$0.25\ \text{kN mm}^{-2} = 0.25 \times \frac{10^3\ \text{N}}{(10^{-3}\ \text{m})^2} = 0.25 \times \frac{10^3}{10^{-6}}\ \text{N m}^{-2} = 2.5 \times 10^8\ \text{N m}^{-2}.$$

Take special care with squared units like mm^2: you must square the factor too.

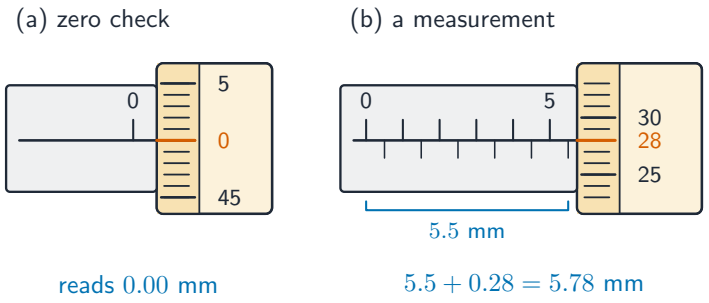


A vernier caliper measures length precisely, with a small uncertainty.

Every **measurement** has some **uncertainty** – we are never fully sure of the value. A good experimenter knows where the uncertainty comes from, makes a fair estimate of it, and carries it through to the final answer.



The main parts of a real micrometer screw gauge, which measures to the nearest 0.01 mm



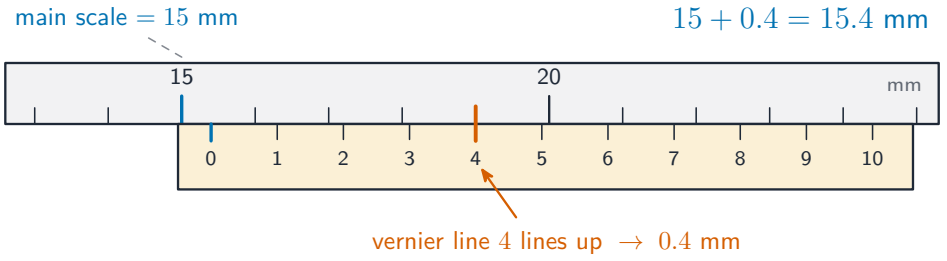
Reading a micrometer: add the main scale reading to the thimble reading



Reading a real micrometer: read the mm and half-mm on the sleeve, then add the thimble scale



Vernier calipers measure to the nearest 0.1 mm – the sliding scale gives the extra digit

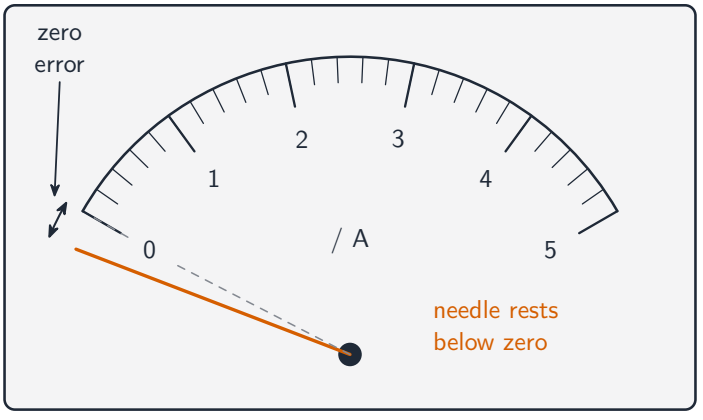


Reading a vernier caliper: whole millimetres from the main scale, plus the tenths from the vernier line that lines up

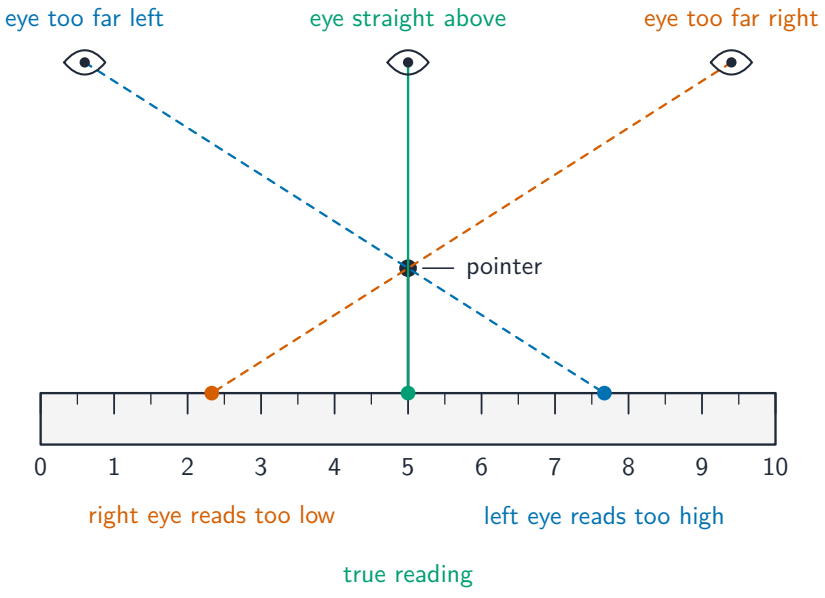
■ Systematic and random errors

A **systematic error** changes every reading by the same amount, in the same direction. You cannot find it by repeating the measurement. Common causes:

- a **zero error** (the scale does not read zero when the true value is zero)
- a **calibration** error (the scale itself is wrong)
- **parallax** (your eye is always to one side of the scale)



An ammeter with a zero error: the needle reads below zero before any current flows



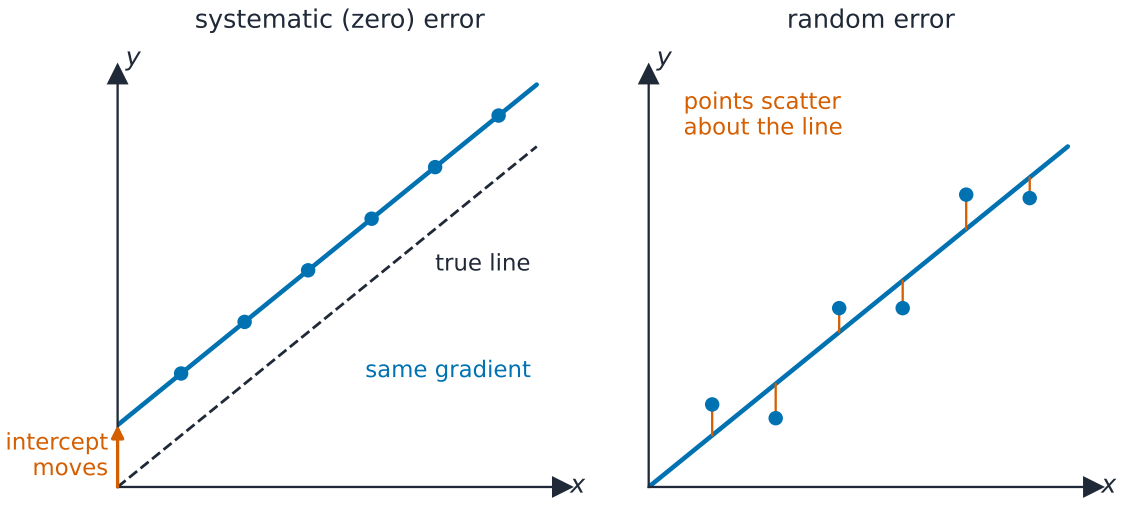
Parallax error: different viewing angles give different scale readings

A systematic error makes the **accuracy** worse, but it does not change the **precision** .

A **random error** makes readings jump above and below the true value, with no pattern. Causes include how carefully you read the scale, changing conditions, and the smallest step the **instrument** can show. If you repeat the measurement many times and take the **mean** (the average), random errors partly cancel out.

A random error makes the precision worse. But with enough repeats, the mean can still be accurate.

On a graph the two errors look different. A systematic error moves every point by the same amount, so the line of best fit keeps its gradient but no longer passes through the origin: the **intercept** changes. Random errors scatter the points above and below the line; a best-fit line drawn through the middle of the scatter still gives a reliable gradient.



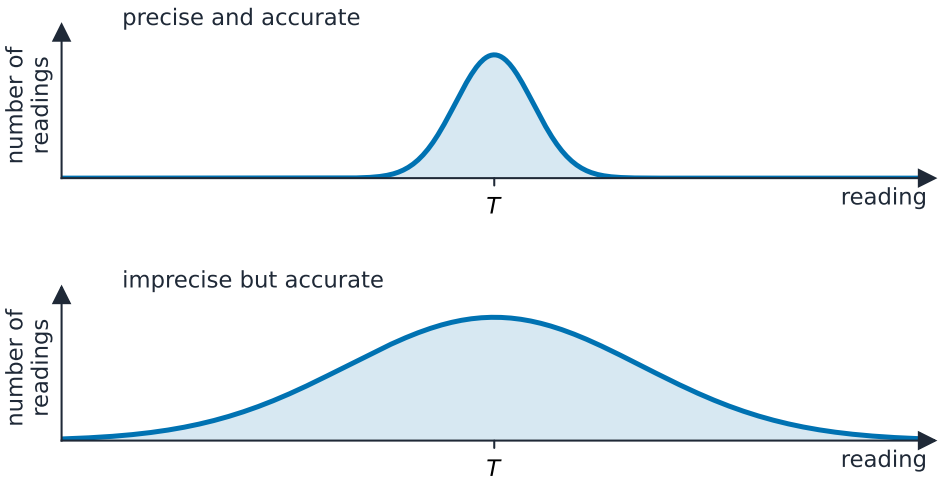
A systematic error shifts the whole line, so the intercept moves; random errors scatter the points about the line

Error	How to reduce it
systematic	check for a zero error and subtract it; calibrate the instrument against a known standard; read the scale from directly in front
random	repeat the reading and take the mean; use an instrument with smaller divisions; time many oscillations instead of one

Precision and accuracy

Precision is how close repeated readings are to each other. Precise readings are grouped very close together.

Accuracy is how close a reading (or the mean of several readings) is to the true value.

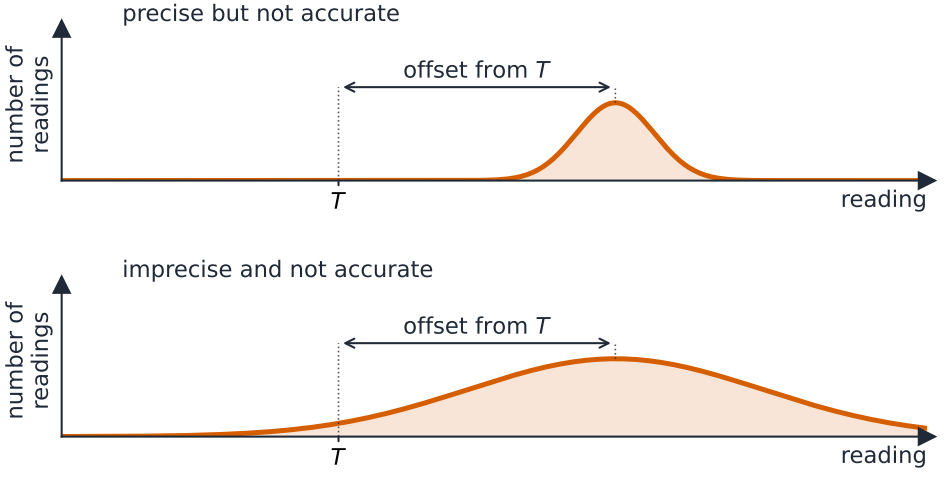


precision: how closely repeated readings agree (random error / spread)

Precision: how narrow the distribution is around the true value T

A set of readings can be:

- precise and accurate – close together and near the true value
- precise but not accurate – close together, but away from the true value (a systematic error)
- accurate but not precise – spread out, but the mean is near the true value
- neither – spread out and away from the true value



accuracy: how close readings are to the true value (systematic error reduces accuracy)

Accuracy: whether the peak of the distribution is centred on the true value T

When a question gives a table of repeated readings, look at the **spread** (precision) and the **mean** (accuracy) separately.

■ Estimating the uncertainty in a reading

Before you combine uncertainties, you need a sensible uncertainty for each raw reading:

- A single reading on an analogue scale: half the smallest **division** (a metre rule reads to 1 mm, so one reading is ±0.5 mm). A length needs two readings, one at each end, so its uncertainty is ±1 mm.
- A digital meter: ±1 in the last digit shown (a stopwatch showing 2.47 s is ±0.01 s).
- A hand-timed measurement: your **reaction time** matters more than the display. Allow about ±0.1 to 0.2 s, and time many oscillations instead of one, so the same uncertainty is shared between all of them.
- Repeated readings: the uncertainty is **half the range** . Three timings of 2.42, 2.48 and 2.45 s have a mean of 2.45 s and a range of 0.06 s, so $t = (2.45 \pm 0.03)$ s.

Be honest. If the edge of a shadow is hard to see, the uncertainty in its position is several millimetres, however fine the scale. Examiners do not accept "the smallest division" as the uncertainty of a difficult reading.

■ Uncertainty in a derived quantity

A measurement is often written as $x \pm \Delta x$. Here Δx is the **absolute uncertainty** . The **percentage uncertainty** is

$$\text{percentage uncertainty in } x = \frac{\Delta x}{|x|} \times 100\%.$$

A **derived quantity** is one you calculate from measured values. Its uncertainty is found by simple rules:

- **Adding or subtracting** – add the absolute uncertainties. If $y = a + b$ or $y = a - b$, then $\Delta y = \Delta a + \Delta b$.
- **Multiplying or dividing** – add the percentage uncertainties. If $y = \dfrac{a \cdot b}{c}$, then

$$\frac{\Delta y}{|y|} = \frac{\Delta a}{|a|} + \frac{\Delta b}{|b|} + \frac{\Delta c}{|c|}.$$

- **Powers** – multiply the percentage uncertainty by the power. If $y = a^n$, then $\dfrac{\Delta y}{|y|} = |n| \cdot \dfrac{\Delta a}{|a|}$.

> **Worked example.** A ball's **diameter** is measured as $d = (5.26 \pm 0.02)$ cm. The **volume** of a sphere is $V = \frac{4}{3}\pi r^3 = \frac{4}{3}\pi(d/2)^3$, so $V \propto d^3$ (V depends on d cubed).

The percentage uncertainty in d is

$$\frac{0.02}{5.26} \times 100\% \approx 0.38\%.$$

Because $V \propto d^3$, the percentage uncertainty in V is three times this, about 1.14%. The volume is $\frac{4}{3}\pi(2.63)^3 \approx 76.2$ cm^3. So the absolute uncertainty is $0.0114 \times 76.2 \approx 0.87$ cm^3. The final answer is $V = (76.2 \pm 0.9)$ cm^3.

Do two values agree? Practical questions often ask whether two calculated values of a constant support a suggested relationship. Do not just say "they are close". Work out the **percentage difference** between them,

$$\text{percentage difference} = \frac{|k_1 - k_2|}{\text{mean of } k_1 \text{ and } k_2} \times 100\%,$$

and compare it with the percentage uncertainty in k (or with the criterion the question gives, often 10%). If the difference is smaller, the two values agree within the uncertainty and the relationship is supported. If it is larger, they do not.

■ Significant figures

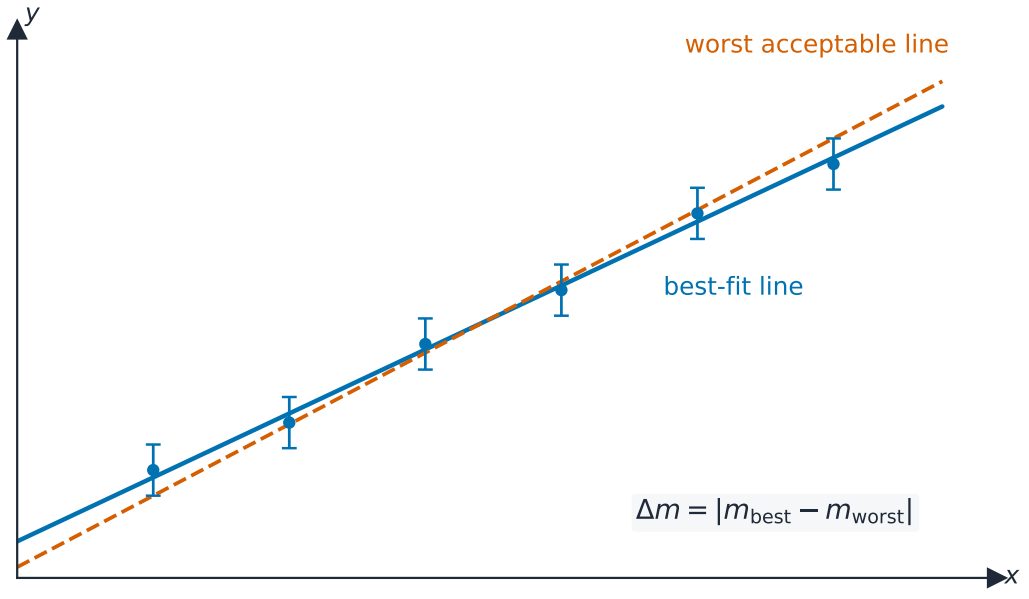
When you write a calculated quantity, give it the **same number of significant figures** as the least precise measurement you used – usually two or three in this syllabus. Too many significant figures makes the answer look more exact than it really is. Too few loses useful information.

Practical papers ask you to **justify** the number you chose. Name the raw readings: "a is given to 2 significant figures because L and t were each measured to 2 significant figures." The phrase "because of the raw data" on its own does not earn the mark. In a table of results, judge each row from the least precise raw reading in that row; do not force every row to the same number of figures.

■ Uncertainties on a graph

Paper 5 asks you to carry uncertainties through a graph:

- Plot each point with an **error bar** whose length shows the absolute uncertainty in that value.
- Draw the **line of best fit** through the points, then the **worst acceptable line**: the steepest (or shallowest) straight line that still passes through every error bar.
- The uncertainty in the gradient is the difference between the two gradients, $\Delta m = |m_{\text{best}} - m_{\text{worst}}|$. The uncertainty in the intercept is found the same way.
- When you plot a logarithm, the absolute uncertainty in $\ln x$ is $\Delta x/x$ (and in $\lg x$ it is $0.434\,\Delta x/x$). Work it out for the largest and smallest values of x separately.



The worst acceptable line still passes through every error bar; the gradient uncertainty is the difference between the two gradients

A percentage uncertainty in the gradient then flows into any quantity you calculate from it, using the rules above.

Scalars and vectors

A **scalar** has size only. A **vector** has both size and direction.

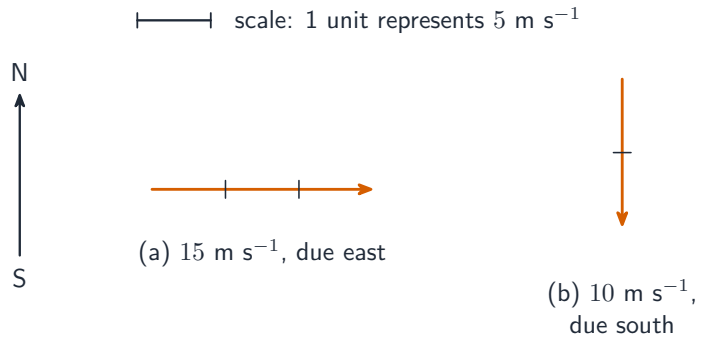
Examples from the syllabus:

- scalars: mass, time, temperature, energy, work, power, distance, speed, pressure, density, **electric charge**
- vectors: **displacement**, velocity, acceleration, force (including weight), **momentum**

Quick test: if it makes sense to ask "in which direction?", the quantity is a vector. You cannot ask "in which direction is the temperature?", so temperature is a scalar. You can ask "in which direction is the velocity?", so velocity is a vector.

Adding and subtracting vectors

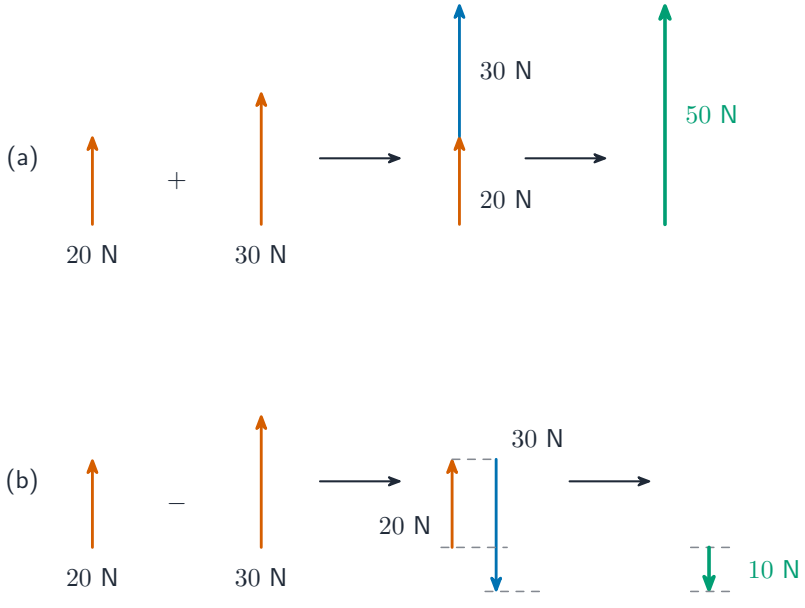
A vector is drawn as an arrow: the **direction** of the arrow gives the direction of the quantity, and the **length** of the arrow (drawn to scale) gives the magnitude.



Vectors represented as arrows drawn to scale

To add two **coplanar** vectors (vectors in the same flat plane), draw them **tip to tail**. The **resultant** goes from the tail of the first arrow to the tip of the second.

To find $\vec{X} - \vec{Y}$, add the reverse of $\vec{Y}$: $\vec{X} + (-\vec{Y})$. The reverse of $\vec{Y}$ has the same size as $\vec{Y}$ but points the opposite way.



Adding and subtracting parallel vectors

If the two vectors are at right angles (90°), the size of the resultant is

$$|\vec{R}| = \sqrt{X^2 + Y^2},$$

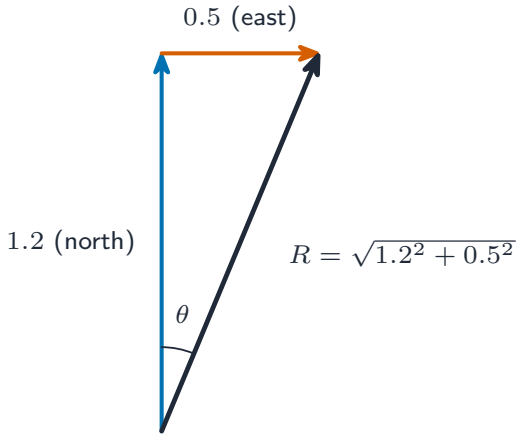
and its direction comes from $\tan\theta = Y/X$.

Worked example. A swimmer heads north at 1.2 m s^{-1} across a river that flows east at 0.5 m s^{-1}. Find the size and direction of the resultant velocity.

The two velocities are perpendicular, so

$$|\vec{R}| = \sqrt{1.2^2 + 0.5^2} = \sqrt{1.69} = 1.3 \text{ m s}^{-1},$$

at an angle $\tan\theta = 0.5/1.2$, giving $\theta \approx 23°$ east of north.



Two perpendicular vectors add to a resultant of size $\sqrt{X^2 + Y^2}$ at angle θ

If two vectors have the same size F with an angle 2α between them, the resultant has size $2F\cos\alpha$ and lies along the line that cuts the angle in half.

Worked example. Two forces of 6.0 N act on an object with 60° between them. Find the resultant.

Here $2\alpha = 60°$, so $\alpha = 30°$ and $R = 2 \times 6.0 \times \cos 30° = 10.4$ N, along the line halfway between the two forces. Resolving gives the same answer: each force has a component $6.0\cos 30°$ along that line, and their components at right angles to it cancel.

For two vectors at any other angle, either make a **scale drawing** of the vector triangle and measure the resultant, or resolve both vectors into perpendicular components, add the components, and recombine with $\sqrt{X^2 + Y^2}$.

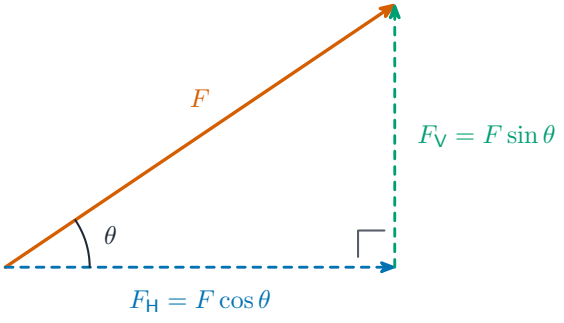
Worked example. A ball moving east at 5.0 m s^{-1} is hit so that it moves north at 5.0 m s^{-1}. Find its change in velocity.

A change in velocity is a vector subtraction: $\Delta\vec{v} = \vec{v}_{\text{final}} - \vec{v}_{\text{initial}}$. Draw 5.0 m s^{-1} north, then add the reverse of the initial velocity, 5.0 m s^{-1} west. The two are perpendicular, so $|\Delta\vec{v}| = \sqrt{5.0^2 + 5.0^2} = 7.1$ m s^{-1}, pointing north-west. The speed did not change, but the velocity did. That is why a force must have acted on the ball (topic 3).

Splitting a vector into perpendicular parts

Any vector can be split into two **perpendicular** (at right angles) **components**. Usually these are **horizontal** and **vertical**, or along and across a surface. For a vector $\vec{v}$ at angle θ to the horizontal:

$$v_{\mathrm{H}} = v\cos\theta, \qquad v_{\mathrm{V}} = v\sin\theta.$$

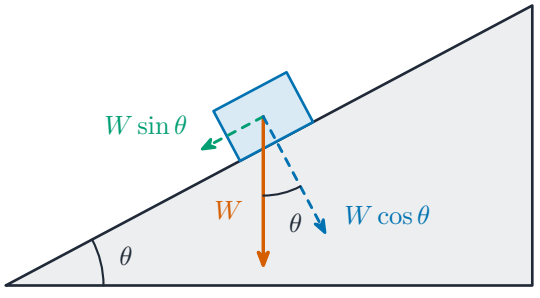


Resolving a vector into horizontal and vertical components

Choose the directions that make the problem easiest. On a slope (an **inclined plane**), split the weight into one part along the slope and one part at right angles to it:

$$W_{\parallel} = W\sin\theta, \qquad W_{\perp} = W\cos\theta,$$

where θ is the angle of the slope to the horizontal.



On a slope the weight splits into $W\sin\theta$ down the slope and $W\cos\theta$ into the slope; the angle between W and the perpendicular is the slope angle θ

You split a vector into components whenever you need to know how much of it acts in one direction. For example: the part of a force that acts along a slope, or the horizontal and vertical parts of a ball's velocity after it is thrown.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
physical quantity	a numerical magnitude together with a unit
homogeneous equation	an equation in which every term has the same base units
systematic error	an error that shifts every reading in the same direction by the same amount; not reduced by repeating
random error	an error that scatters readings above and below the true value; reduced by repeating and averaging
zero error	the reading an instrument shows when the true value is zero
precision	how close repeated readings are to each other
accuracy	how close a reading, or the mean of several readings, is to the true value
absolute uncertainty	the range within which the true value is expected to lie, in the units of the quantity
percentage uncertainty	the absolute uncertainty divided by the value, multiplied by 100%
scalar	a quantity with magnitude only
vector	a quantity with magnitude and direction

Exam tips

- Give every answer a **unit**, and check **homogeneity** – both sides of an equation must have the same base units.
- Distinguish **random error** (reduce by repeating and averaging) from **systematic error** (a zero or calibration error that repeats do not remove).
- Combine uncertainties: **add absolute** uncertainties when adding/subtracting, **add percentage** uncertainties when multiplying/dividing (and multiply the % by any power).
- Distinguish **precision** (small spread) from **accuracy** (close to the true value).
- Resolve a vector into perpendicular components ($F\cos\theta$, $F\sin\theta$); add vectors tip-to-tail or by components.

Common mistakes

- Substituting 5 cm as 5, or 20 g as 20. Convert the prefix first: 0.05 m, 0.020 kg.
- Giving the percentage uncertainty in T^2 as the same as in T. Squaring doubles it; a square root halves it.
- Treating the percentage uncertainty and the absolute uncertainty as the same number.
- Quoting "the smallest division" as the uncertainty of a reading that was hard to take.
- Offering two answers to a definition. Commit to one.

Recognising quantities and estimates

Distinguish the quantity from its unit: temperature is measured in kelvin, time in seconds or minutes, potential difference in volts, and spring constant in N/m. A quantity has a measurable numerical magnitude; a unit name or a category alone does not state a quantity. Values may be fractional and units may be non-SI. Visible light has wavelengths roughly 400–700 nm; green is about 540 nm ($1\ \text{nm} = 10^{-9}\ \text{m}$).

Kinematics

A-Level Physics

Key definitions



A speedometer shows speed: the distance travelled per unit time.

These five quantities come up in almost every **kinematics** question. Learn the exact words – the examiner gives marks for precise wording.

- **distance** – the total length of the path travelled. A **scalar**.
- **displacement** – the straight-line distance from the start to the end, with a direction. A **vector**.
- **speed** – the rate of change of distance with time. A scalar.
- **velocity** – the rate of change of displacement with time. A vector.
- **acceleration** – the rate of change of velocity with time. A vector.

The unit of speed and velocity is m s^{-1}; the unit of acceleration is m s^{-2}.

Two more phrases the examiner uses: **uniform acceleration** means constant acceleration (a straight line on a velocity–time graph), and the **acceleration of free fall** g is the acceleration of an object falling freely in a **uniform gravitational field** when air resistance is **negligible** (small enough to ignore).

A common mistake: **deceleration** just means acceleration in the opposite direction to the velocity. It is not a separate quantity.

Speeds are sometimes given in km h^{-1}. Convert before you calculate: 85 km h^{-1} = $85 \times 1000/3600 = 23.6$ m s^{-1}.



A speed camera measures average speed over a known distance – displacement over time, made into a fine

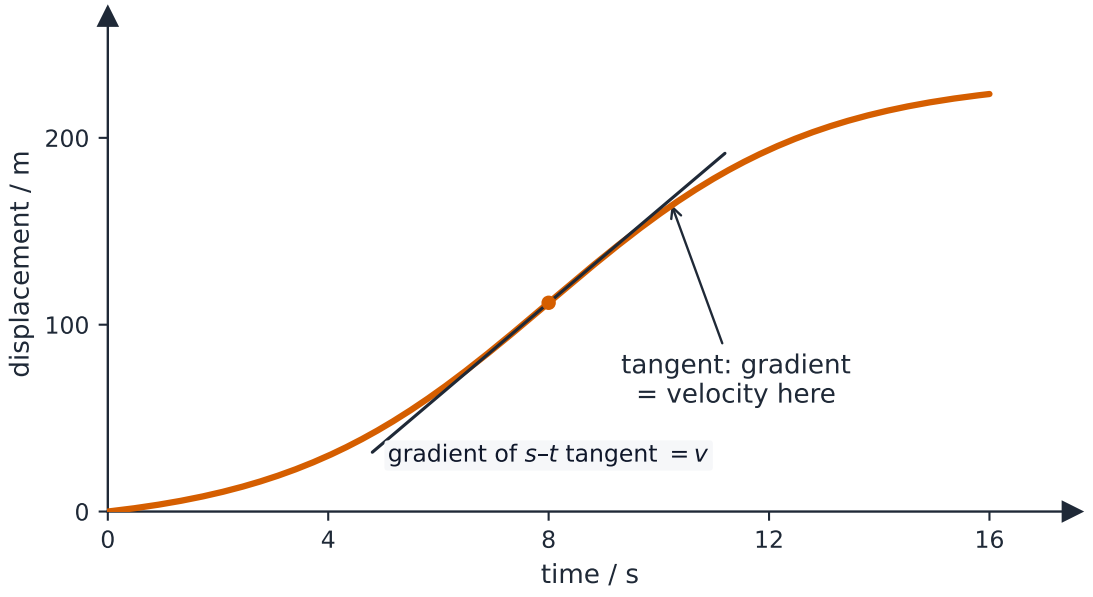
Motion graphs

Many marks come from reading or drawing motion graphs, the **graphical methods** the syllabus names. Two graphs matter.

■ Displacement–time graph

The **gradient** (steepness) of a displacement–time graph at a point gives the velocity at that moment.

- flat line → the object is at rest.
- straight sloping line → constant velocity (gradient = velocity).
- curved line → changing velocity. Draw a **tangent** at the point and find its gradient.



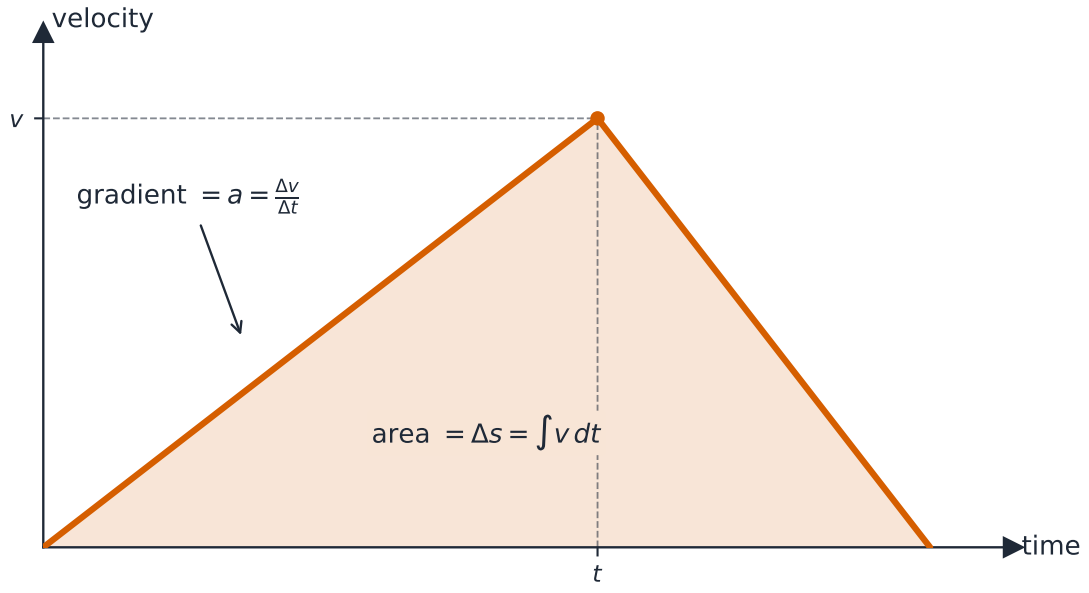
Displacement–time graph of a car on a test track

Velocity–time graph

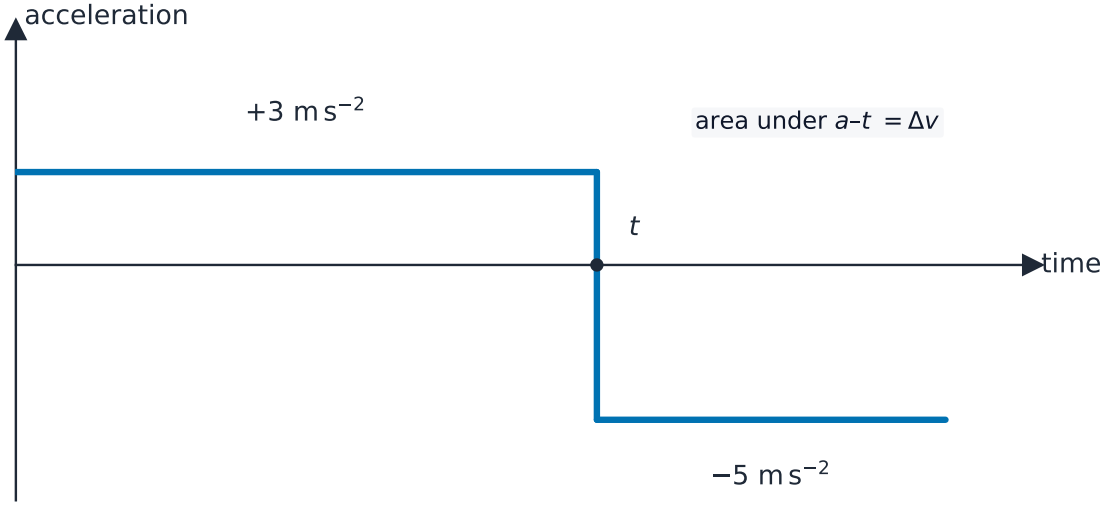
The gradient of a velocity–time graph gives the acceleration at that moment.

The **area** between the line and the time axis gives the displacement in that time.

- flat line → constant velocity (zero acceleration).
- straight sloping line → uniform acceleration (constant acceleration).
- curved line → changing acceleration.
- area above the time axis is positive displacement; area below is negative (the object moved backwards).



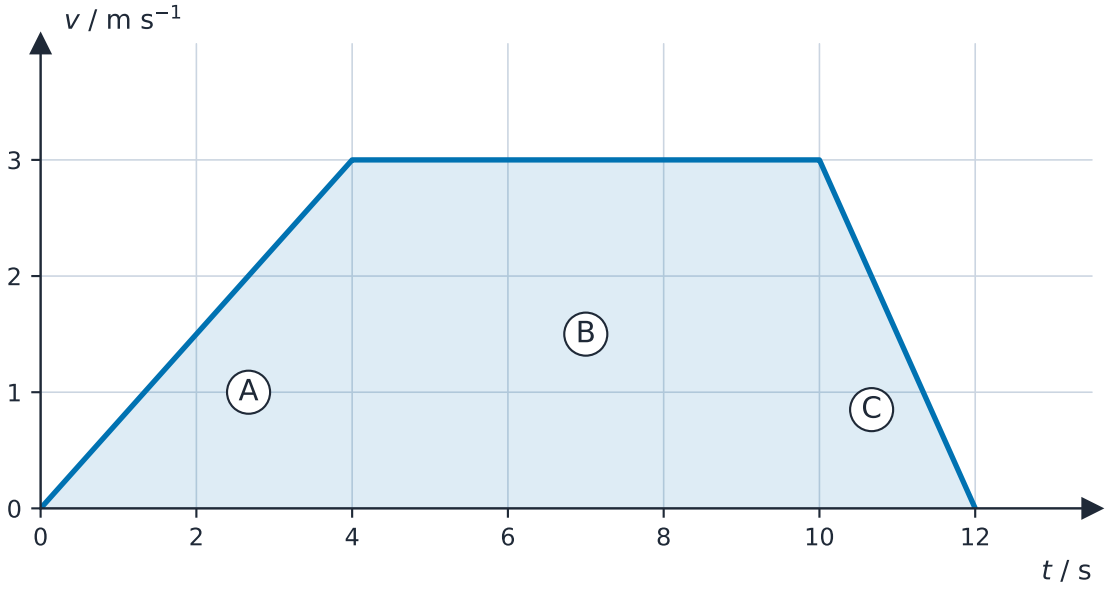
Velocity–time graph – gradient gives acceleration, area gives displacement



Acceleration–time graph derived from the same motion

To find the displacement, split the area into triangles and rectangles, or count grid squares. Area of a triangle is $\frac{1}{2} \times$ base $\times$ height; area of a rectangle is base $\times$ height.

> **Worked example.** The graph shows the velocity of a lift. Find the acceleration in each stage and the total distance travelled.



Velocity–time graph of a lift: gradient gives each acceleration, the shaded area gives the distance

- Stage A: gradient $= \dfrac{3.0 - 0}{4.0} = 0.75$ m s^{-2}.
- Stage B: the line is flat, so the acceleration is zero.
- Stage C: gradient $= \dfrac{0 - 3.0}{2.0} = -1.5$ m s^{-2} (a deceleration of 1.5 m s^{-2}).
- Distance: triangle A $= \frac{1}{2} \times 4.0 \times 3.0 = 6.0$ m, rectangle B $= 6.0 \times 3.0 = 18$ m, triangle C $= \frac{1}{2} \times 2.0 \times 3.0 = 3.0$ m; total 27 m.

When the graph is a curve, the area is still the displacement: count the squares under the curve (half a square or more counts as one) and multiply by the value of one square.

The four SUVAT equations

For motion in a straight line with uniform acceleration, we use five symbols: starting velocity u, final velocity v, acceleration a, displacement s, and time t. Four equations link them:

$$v = u + at$$

$$s = ut + \tfrac{1}{2}at^2$$

$$s = \tfrac{1}{2}(u + v)t$$

$$v^2 = u^2 + 2as$$

Each equation uses four of the five symbols. To pick the right one: write down what you know and what you want, then choose the equation with exactly those four.

> **Worked example.** A car accelerates uniformly from 8 m s^{-1} to 20 m s^{-1} over a distance of 56 m. Find its acceleration.

We know u, v and s and want a, so use $v^2 = u^2 + 2as$:

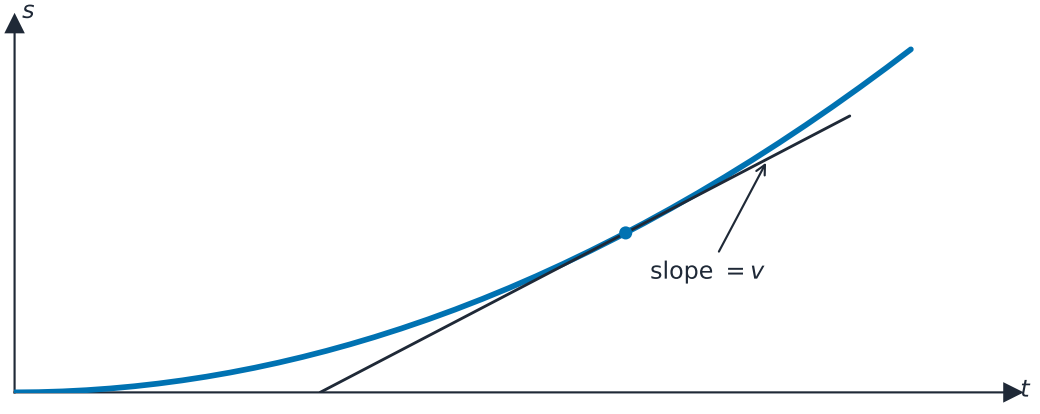
$$20^2 = 8^2 + 2a(56) \quad \Rightarrow \quad 336 = 112\,a \quad \Rightarrow \quad a = 3.0 \text{ m s}^{-2}.$$

Where the SUVAT equations come from

You should be able to get these from the definitions of velocity and acceleration. Think of the velocity–time graph of the motion: a straight line from u at $t = 0$ to v at time t.

- $v = u + at$: acceleration is the gradient, $a = (v - u)/t$, so $v = u + at$.
- $s = \frac{1}{2}(u + v)t$: displacement is the area under the line, a **trapezium** with parallel sides u and v and width t, so $s = \frac{1}{2}(u + v)t$.
- $s = ut + \frac{1}{2}at^2$: put $v = u + at$ into the area: $s = \frac{1}{2}(u + u + at)t = ut + \frac{1}{2}at^2$.
- $v^2 = u^2 + 2as$: from the first equation $t = (v - u)/a$; put this into $s = \frac{1}{2}(u + v)t$ to get $2as = (v + u)(v - u) = v^2 - u^2$.

A "derive" question wants exactly these steps, each starting from a definition or an equation already established, with no numbers.



constant $a \Rightarrow s$–t is a parabola · gradient of tangent $= v$

Displacement–time graph for uniform acceleration – the slope at any point equals the instantaneous velocity

If a question asks "which equation can be found using only the gradient of a velocity–time graph?", the answer is $v = u + at$ (the gradient is the acceleration).

"Show that" questions. When the question gives the answer ("show that the height is 3.2 km"), the marks are for the method. State the equation, substitute every value with its unit, and give the result to one more significant figure than the value quoted (3.24 km, which rounds to 3.2 km). Reaching the quoted value proves nothing on its own.

■ Choosing a positive direction

Pick a positive direction at the start and keep it. Anything pointing the other way gets a minus sign. For a ball thrown straight up, if "up" is positive: u is positive, $a = -g$ (**gravity** pulls down), and at the highest point the displacement is positive but the velocity is zero.

Free fall under gravity

When **air resistance** can be ignored, an object in **free fall** has a constant acceleration $g \approx 9.81$ m s^{-2} downwards. This is the same for every mass.

For a ball dropped from rest and falling a distance h:

$$h = \tfrac{1}{2}gt^2, \qquad v = gt, \qquad v^2 = 2gh.$$

For a ball thrown straight up with speed u:

- greatest height: put $v = 0$ in $v^2 = u^2 - 2gh$, giving $h = u^2/(2g)$.
- time to reach the top: put $v = 0$ in $v = u - gt$, giving $t = u/g$.
- total time to fall back to the start height: $2u/g$ (the motion is **symmetric**).

> **Worked example.** A ball is thrown straight up at 20 m s^{-1}. Find the greatest height it reaches (take $g = 9.81$ m s^{-2}).

At the highest point $v = 0$, so from $h = u^2/(2g)$:

$$h = \frac{20^2}{2 \times 9.81} = \frac{400}{19.62} \approx 20.4 \text{ m}.$$

■ When air resistance is not negligible

The equations above assume the only force is the weight. With air resistance, the resultant force on a falling object is smaller than its weight, so the acceleration is less than g; and because air resistance grows with speed, the acceleration keeps decreasing as the object speeds up. On a velocity–time graph the line curves, its gradient falling towards zero as the object approaches **terminal velocity** (topic 3). For a projectile, air resistance shortens the range and lowers the maximum height, and the path is no longer a symmetrical **parabola**: the object comes down more

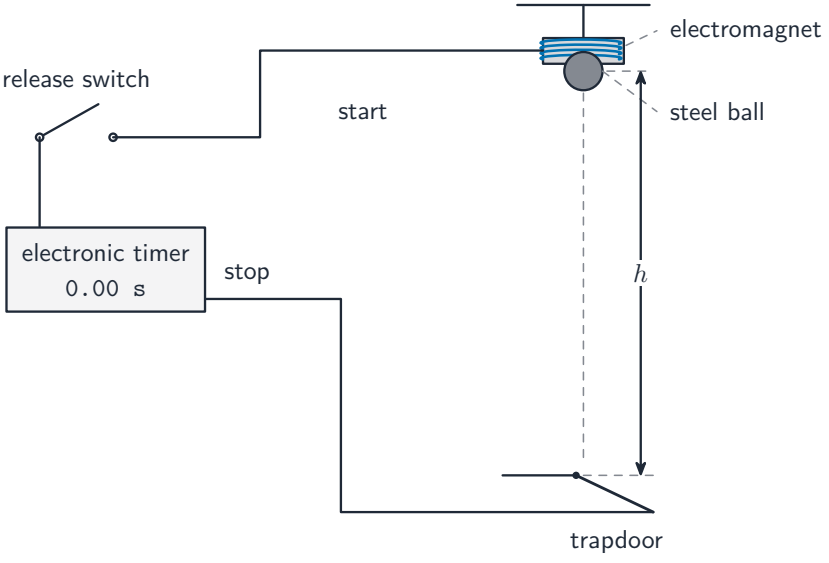
steeply than it went up. A "state and explain" question wants the force first, then its effect on the acceleration, then the effect on the motion.

■ Experiment to find g

A common method: drop an object from rest, then measure the distance h it falls and the time t it takes. Then

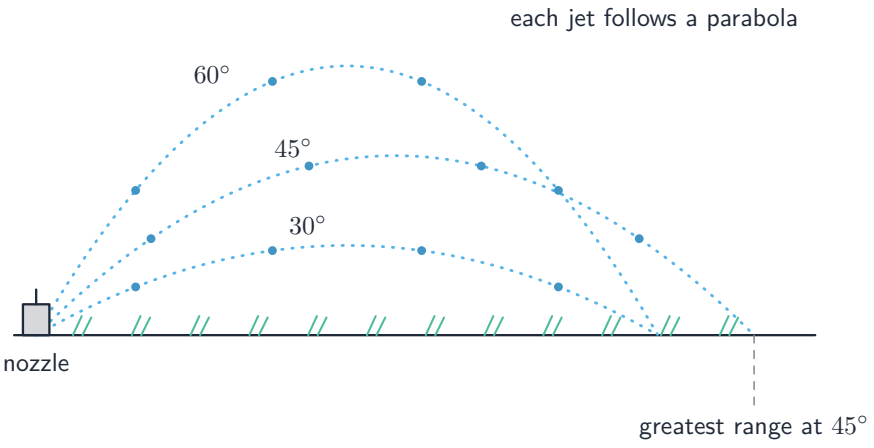
$$g = \frac{2h}{t^2}.$$

Repeat for several heights and plot h against t^2. The gradient of the best straight line is $g/2$, so g is twice the gradient. Repeating reduces **random error** . An electronic timer – using **light gates** , or a switch the ball hits – removes **reaction-time** error.



Experimental set-up for measuring the acceleration due to free fall

Motion in two directions



Water jets from a sprinkler trace parabola paths – a real example of projectile motion

When an object moves at constant velocity in one direction (say **horizontal**) and speeds up in a direction at right angles to it (say **vertical** , under gravity), the two motions do not affect each other. Treat each direction on its own, with its own SUVAT equation.

Horizontal throw

An object thrown horizontally with speed u_{H} from height h, with air resistance ignored:

- **horizontal**: constant velocity u_{H}. After time t, the horizontal distance is $x = u_{\mathrm{H}}t$.
- **vertical**: starts from rest and speeds up downwards at g. After time t, it has fallen $y = \frac{1}{2}gt^2$ and has vertical velocity $v_{\mathrm{V}} = gt$.

The time to reach the ground depends **only** on the height h, not on u_{H}. Solve $h = \frac{1}{2}gt^2$ for t; then the horizontal **range** is $u_{\mathrm{H}}t$.

> **Worked example.** A ball is thrown horizontally at 15 m s^{-1} from the top of a cliff 20 m high. Find the time it takes to land and how far from the base it lands (take $g = 9.81$ m s^{-2}).

Vertical motion gives the time: from $h = \frac{1}{2}gt^2$,

$$t = \sqrt{\frac{2h}{g}} = \sqrt{\frac{2 \times 20}{9.81}} \approx 2.0 \text{ s}.$$

Horizontal motion then gives the range: $x = u_{\mathrm{H}}t = 15 \times 2.0 \approx 30$ m.

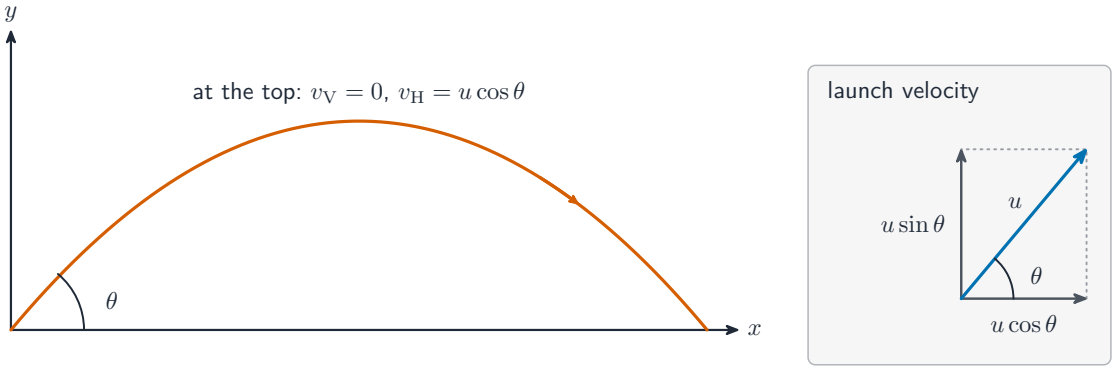
The horizontal-velocity graph is a flat line at u_{H}. The vertical-velocity graph is a straight line from the origin with gradient g.

Comparing two objects. If object A is dropped and object B is thrown horizontally from the same height at the same moment, both reach the ground at the same time: their vertical motions are identical. If instead B is thrown with an upward component,

it takes longer, because it must first rise and come back to the start height before it falls the same distance. Its speed on landing is still found from energy: the kinetic energy gained equals the potential energy lost, whatever the direction of the throw, so a ball launched at any angle with the same speed from the same height lands at the same speed (topic 5).

■ Projectile at an angle

A **projectile** thrown at speed u at angle θ above the horizontal:

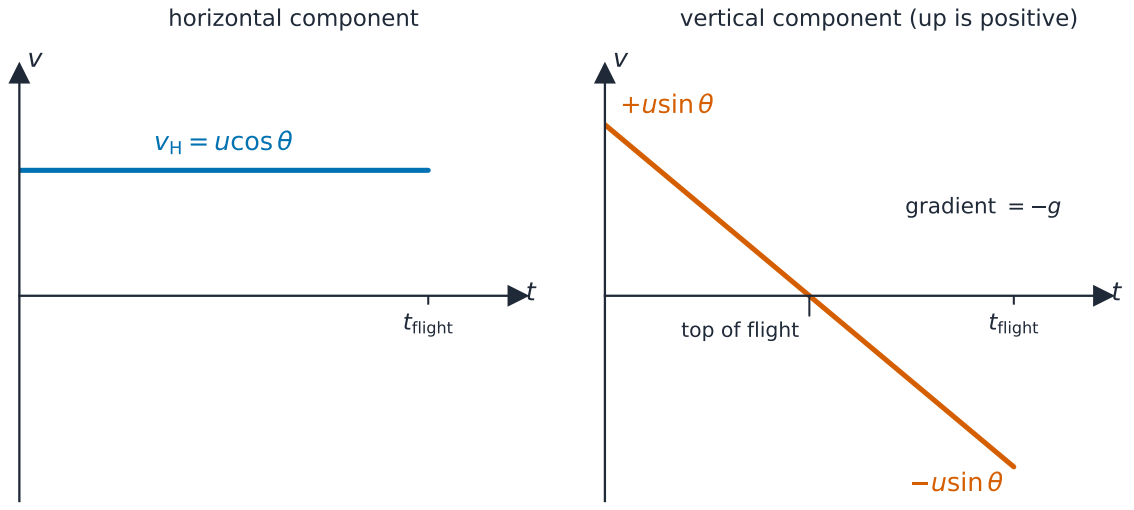


Projectile launched at angle θ – horizontal and vertical motions are independent

- horizontal **component** of the starting velocity: $u_H = u\cos\theta$ (stays constant during the flight).
- vertical component of the starting velocity: $u_V = u\sin\theta$ (gets smaller, becomes zero at the top, then grows downwards).

At the highest point, $v_V = 0$, but v_H is still $u\cos\theta$. The time to the top is $t_{up} = u\sin\theta/g$; the total flight time (back to the start height) is $2t_{up}$.

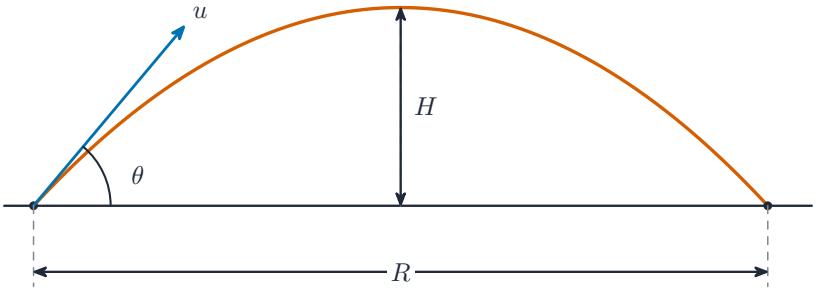
Sketching the velocity graphs. Examiners often ask you to sketch v_H and v_V against time on the same axes, taking upwards as positive. The horizontal component is a flat line at $u\cos\theta$ for the whole flight. The vertical component is a straight line of gradient $-g$: it starts at $+u\sin\theta$, crosses zero at the top of the flight, and reaches $-u\sin\theta$ on landing at the start height. Label each line, mark the times where the lines start and stop, and make the crossing sit exactly at t_{up}.



Velocity–time sketches for a projectile: the horizontal component stays constant; the vertical component falls in a straight line through zero at the top

Worked example. A ball leaves the ground at 18 m s^{-1} at 60° to the horizontal. Show that it reaches its maximum height at $t = 1.6$ s.

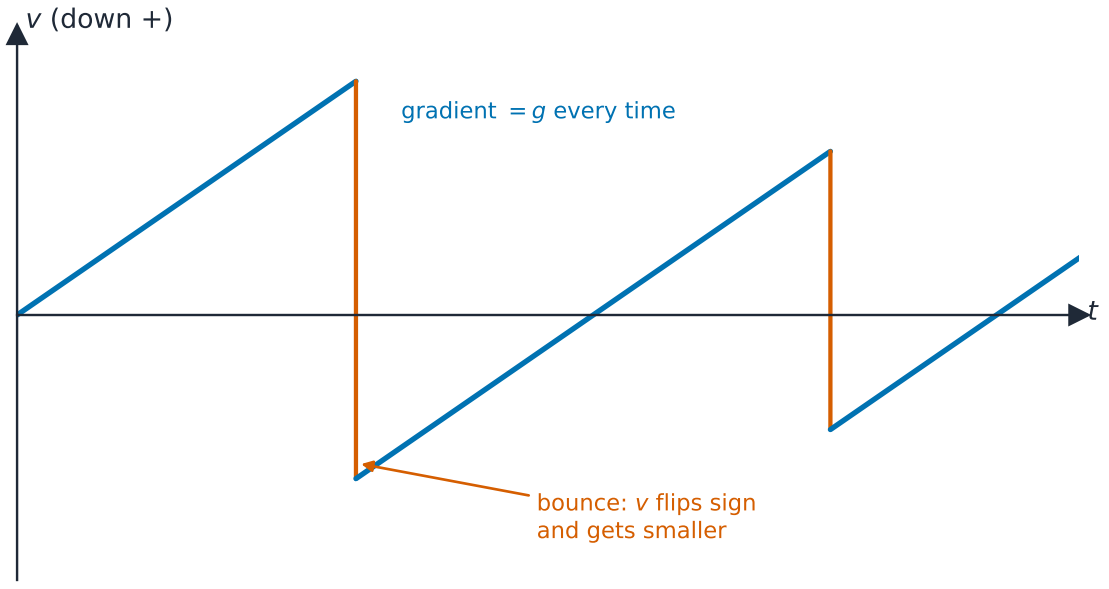
$u_V = 18\sin 60° = 15.6$ m s^{-1}. At the top $v_V = 0$, so from $v = u + at$ with $a = -9.81$ m s^{-2}: $t = 15.6/9.81 = 1.59$ s ≈ 1.6 s.



Range R of a projectile launched from and landing on level ground

Bouncing ball

When a ball bounces, its velocity–time graph is a set of straight sloping lines (constant g) with a sudden jump at each bounce (the velocity flips direction, and gets smaller if some **energy** is lost). Add up the times and the distances across the bounces.



A bouncing ball: every sloping line has gradient g; the velocity flips sign and shrinks at each bounce

Read such a graph carefully: the gradient of every sloping line is the same g; the ball is at its highest point wherever the line crosses the time axis; and the height of each bounce comes from the area of the triangle above (or below) the axis for that flight.

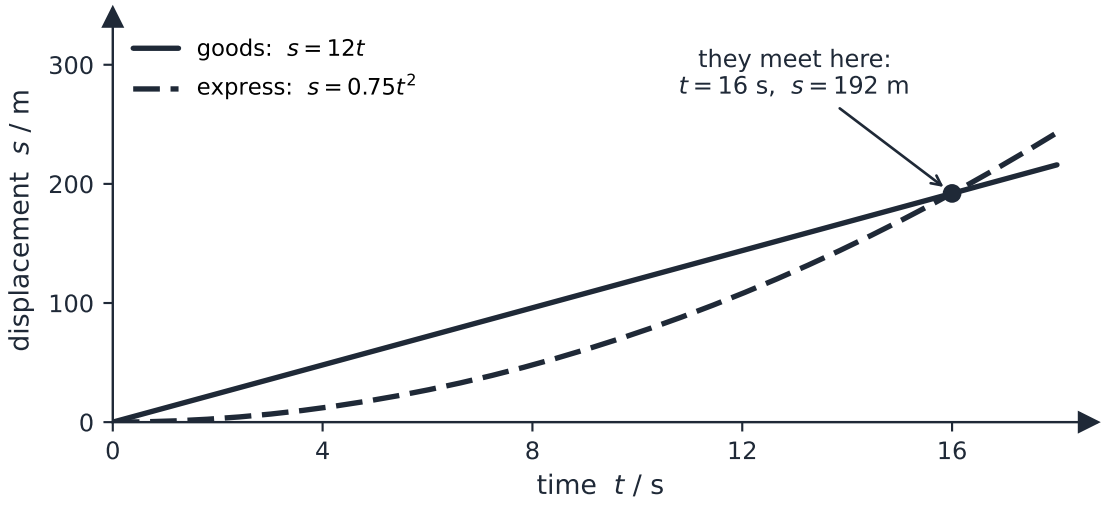
Two objects meeting

When two objects move along the same line in different ways, write a displacement equation for each. Use the same start time and the same positive direction. Then set the two displacements equal (or set their difference to a given gap).

For a goods train at constant velocity u_{G} and an express train starting from rest with acceleration a, both passing the same point at $t = 0$:

$$s_{\mathrm{G}} = u_{\mathrm{G}} t, \qquad s_{\mathrm{E}} = \tfrac{1}{2} a t^2.$$

They are level again when $s_{\mathrm{G}} = s_{\mathrm{E}}$, giving $t = 2u_{\mathrm{G}}/a$.



Meet where the displacement–time graphs cross (same s at same t)

Tips for solving problems

1. **Draw a diagram** and mark the positive direction.
2. **List the SUVAT symbols** with their known and unknown values, including signs.
3. **Choose the SUVAT equation** with exactly the four symbols you have, plus the one you want.
4. For projectile motion, **split into horizontal and vertical** SUVAT problems, linked only by the time t.
5. Always **check the units** of your answer, and that its size is sensible.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
distance	the total length of the path travelled (a scalar)
displacement	the distance moved in a stated direction, from start to finish (a vector)
speed	the rate of change of distance with time
velocity	the rate of change of displacement with time
acceleration	the rate of change of velocity with time
uniform acceleration	acceleration that is constant in magnitude and direction
acceleration of free fall	the acceleration of an object falling freely under gravity alone, with air resistance negligible

Exam tips

- Use the **SUVAT** equations only for **constant acceleration**; list s, u, v, a, t and pick the equation missing your unknown.
- Choose one direction as **positive** and keep signs consistent (usually $g = -9.81$ m s^{-2}).
- On a **velocity-time graph**, gradient = acceleration and area = displacement.
- For projectiles, treat **horizontal (constant velocity) and vertical ($a = g$) motion separately**, linked by the same time.

Common mistakes

- Saying the heavier ball hits the ground first, or faster, when air resistance is ignored. Both fall with the same acceleration and land at the same speed.
- Taking the gradient of a v–t graph when the question wants the area, or the area when it wants the gradient. Gradient is acceleration; area is displacement.
- Drawing a curve on a v–t graph for a body under constant acceleration. Constant acceleration is a straight line; only the sign of v changes at a bounce.
- Using $g = +9.81$ for a ball thrown upwards with "up" as positive. Once up is positive, $a = -9.81$ m s^{-2} for the whole flight, on the way down too.
- Using a SUVAT equation when the acceleration is changing (air resistance, a curved graph). Then only the graph methods work.

Dynamics

A-Level Physics

Mass, momentum and force

Mass

Mass tells you how hard it is to change an object's motion. The larger the mass, the larger the **force** needed to give it a certain **acceleration**. Mass is measured in kilograms (kg) and is a **scalar**.

Momentum

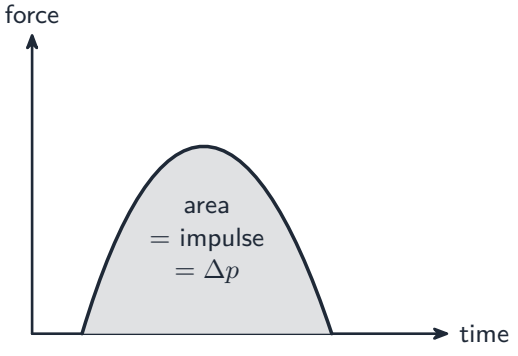
Linear momentum is the product of mass and velocity:

$$p = mv.$$

Momentum is a **vector** – it points the same way as the **velocity**. Its unit is kg m s^{-1}, which is the same as N s.

Force as the rate of change of momentum

Newton's second law, in its general form: the **resultant force** on an object equals the rate of change of its momentum.



Impulse is the area under a force-time graph, equal to the change in momentum

$$F = \frac{\Delta p}{\Delta t}.$$

When the mass is constant this becomes $F = ma$, because $\Delta p = m\,\Delta v$ and $\Delta v/\Delta t = a$. Cambridge questions often want you to use $F = \Delta p/\Delta t$ directly for a **collision** or an **impulse**: the average force equals the change in momentum divided by the contact time.

A ball hits a wall with momentum p_1 and bounces back with momentum p_2 in the

opposite direction. The change in momentum is $\Delta p = p_2 - p_1$ (give each direction the correct sign). The average force is $\Delta p / \Delta t$, where Δt is the contact time.

> **Worked example.** A 0.20 kg ball hits a wall at 8.0 m s^{-1} and bounces straight back at 6.0 m s^{-1}. The contact lasts 0.050 s. Find the average force on the ball.

Take the rebound direction as positive, so $u = -8.0$ m s^{-1} and $v = +6.0$ m s^{-1}:

$$\Delta p = m(v - u) = 0.20 \times \big(6.0 - (-8.0)\big) = 2.8 \text{ kg m s}^{-1},$$

$$F = \frac{\Delta p}{\Delta t} = \frac{2.8}{0.050} = 56 \text{ N}.$$

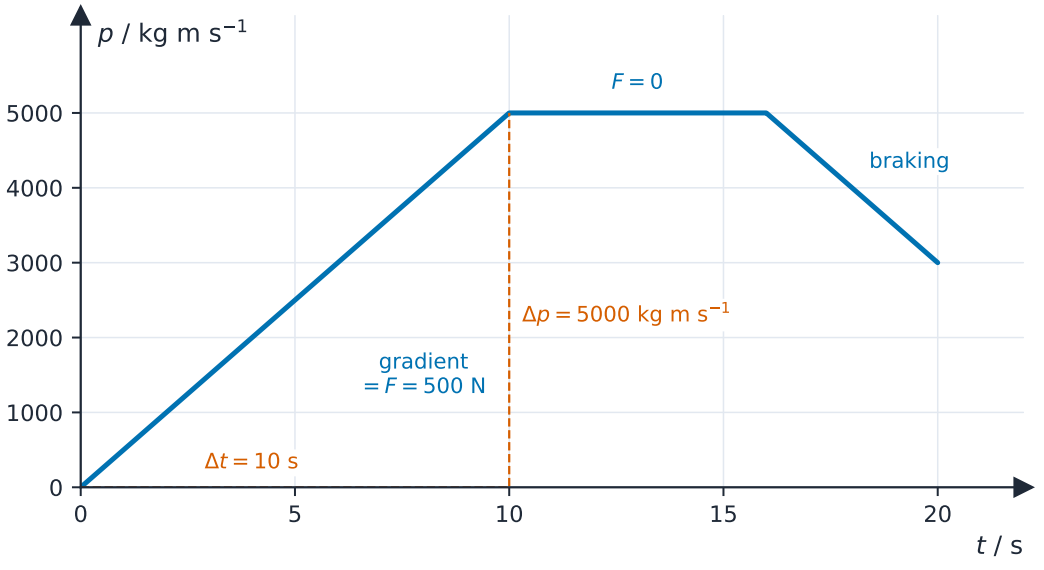
When you know the momentum but not the speed, the change in **kinetic energy** is

$$\Delta E_\text{k} = \frac{p_2^2 - p_1^2}{2m}.$$

This comes from $E_\text{k} = \tfrac{1}{2}mv^2 = p^2/(2m)$.

Momentum–time graphs

Because $F = \Delta p / \Delta t$, the **gradient** of a momentum–time graph is the resultant force, just as the gradient of a velocity–time graph is the acceleration. A straight line means a constant resultant force; a horizontal line means zero resultant force; a line sloping down means a force acting against the motion.



The gradient of a momentum–time graph is the resultant force: 500 N while accelerating, zero while the momentum is constant, negative while braking

Worked example. The graph shows the momentum of a motorcycle. Find the resultant force on it during the first 10 s.

Gradient $= \Delta p/\Delta t = 5000/10 = 500$ N. Between 10 s and 16 s the momentum is constant, so the resultant force is zero: the driving force just balances the resistive forces. From 16 s to 20 s the momentum falls by 2000 kg m s^{-1}, so the resultant force is -500 N, acting backwards.

If a question tells you the speed is changing while the resultant force stays constant, that force cannot be air resistance: drag depends on speed, so it would change as the speed changes, and drag always acts against the motion.

Newton's three laws of motion



A rocket pushes gas down; by Newton's third law the gas pushes the rocket up.

First law

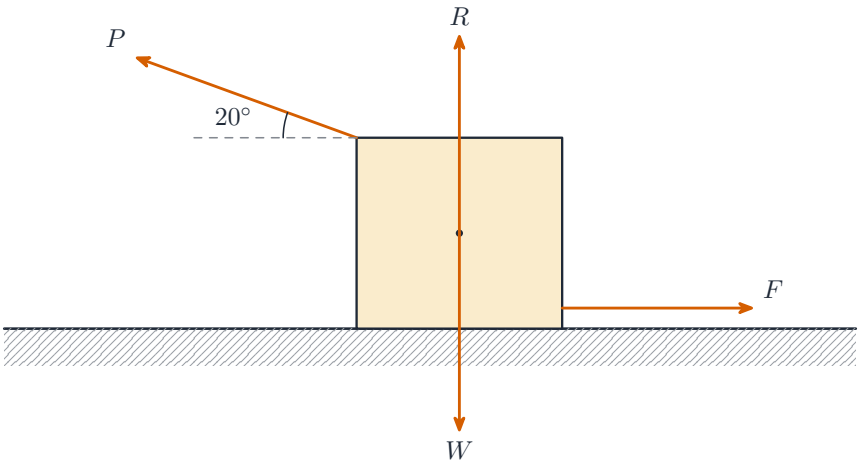
An object stays at rest, or keeps moving at constant velocity in a straight line, unless a resultant **external force** acts on it. In short: zero resultant force means zero acceleration.

Second law

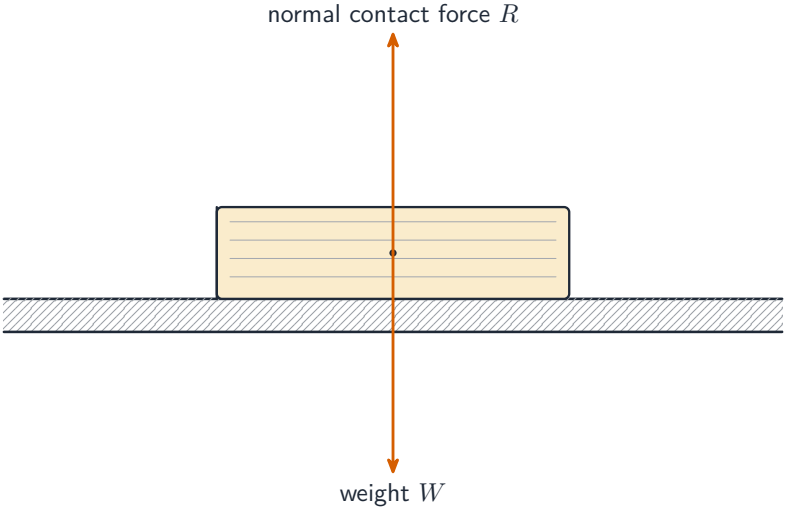
The resultant force on an object equals its rate of change of momentum, and acts in the same direction as that change. In SI units,

$$F = \frac{\Delta p}{\Delta t} = ma \quad \text{(for constant mass)}.$$

Acceleration and resultant force always point the **same way**.



Free-body diagram showing all forces on a block being pulled at an angle



Weight and normal contact force on a book resting on a table

Third law

When body A pushes on body B, body B pushes back on body A with an **equal and opposite** force. The two forces:

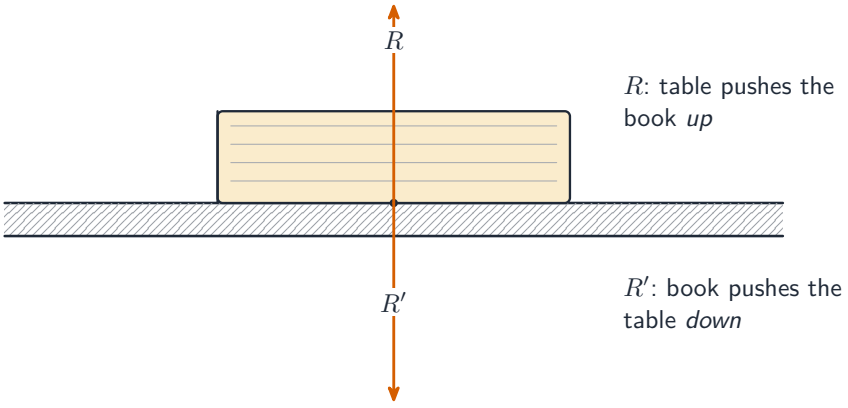
- act on **different objects**,
- are of the **same type** (both gravitational, both contact, both **electrostatic** , and so on),
- have the same size and opposite direction.

A common trap: the **weight** of a block on a table and the **normal contact force** from the table are **not** a third-law pair (they act on the same object and

are different types). The third-law partner of the block's weight is the pull the block makes on the Earth. The third-law partner of the table's contact force is the push the block makes on the table.

To name a third-law partner, swap the two bodies in the sentence. "The back wheel of a bicycle pushes backwards on the road" pairs with "the road pushes forwards on the back wheel", and that forward push is what drives the bicycle. "The Earth pulls the Moon" pairs with "the Moon pulls the Earth", with the same size of force.

For a rocket: the **thrust** on the rocket and the force on the gases are a third-law pair (the engine pushes the gas down, the gas pushes the engine up). Weight and air resistance are not part of this pair.



Newton's third-law pair: R on the book (up) and R' on the table (down)

Weight

Weight is the force on an object from a **gravitational field** . Near the Earth's surface,

$$W = mg,$$

where $g \approx 9.81$ m s^{-2} is the acceleration of free fall. Weight is a vector that points towards the centre of the Earth. Do not mix it up with mass: mass is the same everywhere, but weight changes with place.

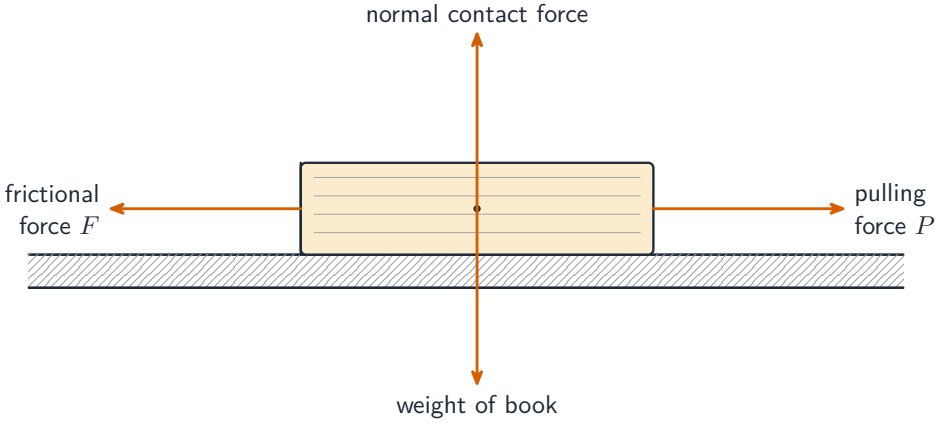
On the Moon g is about 1.6 m s^{-2}, so an astronaut's mass is unchanged but their weight is about one sixth of its value on Earth. When a multiple-choice question asks which statement describes weight, the answer is the one that says "the force on the object due to a gravitational field", not "mass times acceleration" and not "the resultant force".

Non-uniform motion: friction, drag and terminal velocity

Friction and drag forces

A **friction** force between two solid surfaces acts along the surface and opposes the sliding. A **viscous** or **drag** force is the resistive force from a **fluid** (a liquid or gas) on an object moving through it; air resistance is the case for air. You do not need to use any **coefficient** of friction or viscosity.

A simple model: the drag gets bigger as the speed gets bigger. At zero speed, the drag is zero. Drag also grows with the **cross-sectional area** of the object and with the density of the fluid; a parachute works by making the area, and so the drag, much larger. These frictional forces and drag forces are all resistive forces: they act against the direction of motion and transfer kinetic energy to thermal energy.



Free-body diagram of a book being pulled on a table

An object falling through air

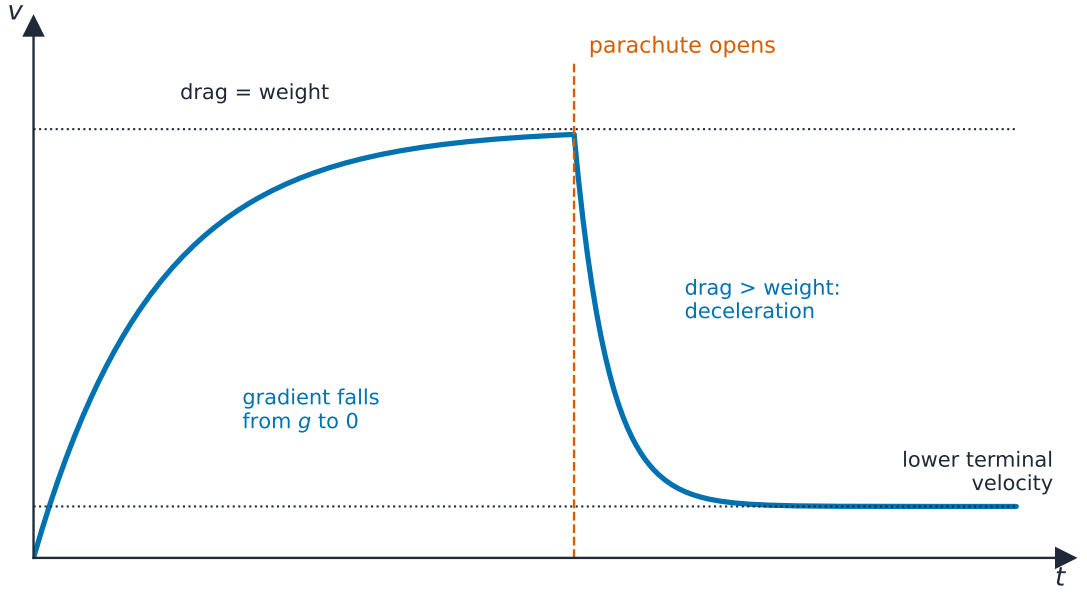
For an object dropped from rest and falling through air:

1. At first, only weight acts, so the object speeds up downwards at g.
2. As the speed grows, the upward drag grows. The resultant force gets smaller, so the acceleration gets smaller.
3. In the end, the drag equals the weight. The resultant force is zero, the acceleration is zero, and the speed stays constant – the **terminal velocity** .

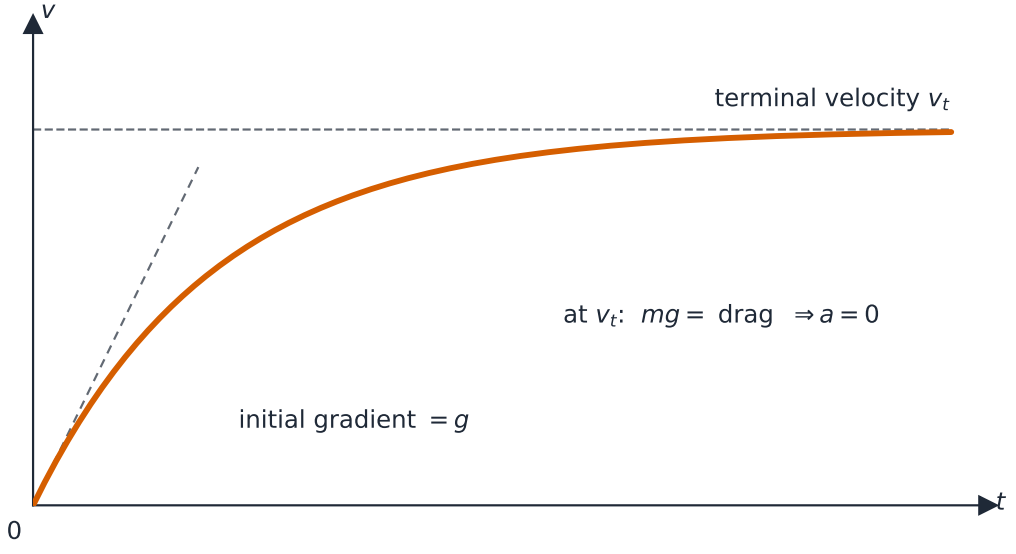
On a velocity–time graph, the line starts straight with gradient g, then bends and flattens at the terminal velocity. This shape (fast start, then slowing acceleration, then constant speed) is how "falling with air resistance" differs from "free fall in a vacuum" in a **uniform gravitational field** . On an acceleration–time graph the acceleration starts at g and decreases, curving down to zero.

Two spheres of the same size but different density have different terminal velocities: the denser sphere has the larger weight, so it needs a larger drag to balance it, and

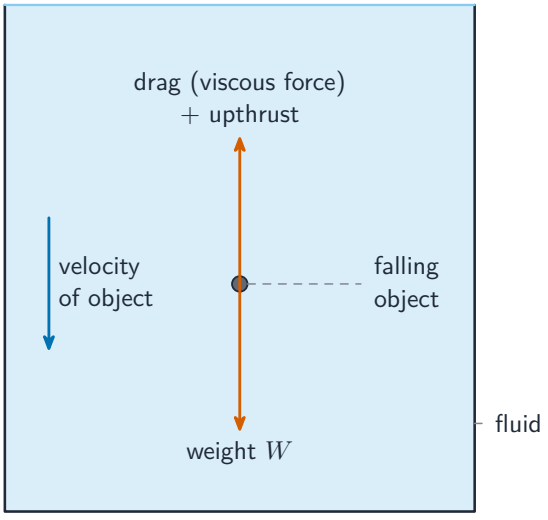
drag only reaches that value at a higher speed. That is also why a parachute changes everything: when it opens, the drag suddenly exceeds the weight, the resultant force is upwards, and the skydiver **decelerates** (the acceleration is upwards while the velocity is still downwards) until the drag has fallen back to equal the weight at a new, much lower terminal velocity.



A skydiver's velocity–time graph: the acceleration falls from g to zero at the first terminal velocity; opening the parachute makes the drag exceed the weight, so the velocity falls to a new, lower terminal velocity



Velocity–time graph for an object falling through air



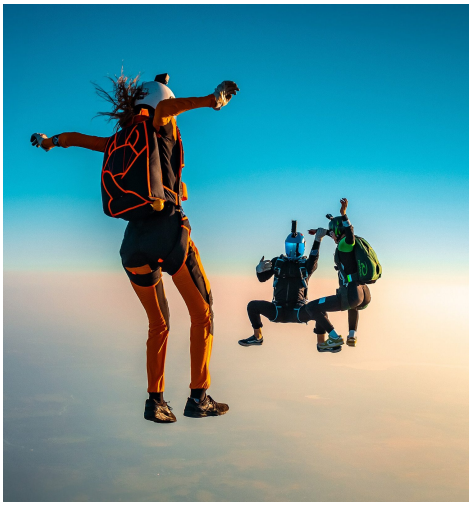
Forces on a falling object in a fluid

Falling through a liquid: three forces

A ball falling through a liquid has three forces on it: its weight downwards, the **upthrust** upwards, and the **viscous drag** upwards. The upthrust comes from **hydrostatic pressure** : the pressure in a liquid increases with depth, so the liquid pushes harder on the bottom of the ball than on the top, and the resultant of these pushes is upwards. Its size is the weight of the liquid the ball displaces, $U = \rho_{\text{liquid}} V g$ (topic 4). The drag grows with speed. When weight = upthrust + drag, the resultant force is zero and the ball falls at its terminal speed. A "draw labelled arrows" question wants exactly these three, with the upthrust and drag both up and the weight down.

> **Worked example.** A steel ball of radius 2.0 mm and weight 2.56×10^{-3} N falls through oil of density 850 kg m^{-3}. The viscous drag on it is $F = 6\pi\eta r v$, where η is the **viscosity** of the oil and v the speed. (a) Find the SI base units of η. (b) Show that the upthrust on the ball is 2.8×10^{-4} N. (c) Taking $\eta = 0.20$ in SI units, find the terminal speed.

(a) $\eta = F/(6\pi r v)$, so its units are $\text{N}/(\text{m} \cdot \text{m s}^{-1}) = \text{kg m s}^{-2}/(\text{m}^2\ \text{s}^{-1}) = \text{kg m}^{-1}\ \text{s}^{-1}$.
(b) $V = \frac{4}{3}\pi r^3 = \frac{4}{3}\pi(2.0 \times 10^{-3})^3 = 3.35 \times 10^{-8}$ m^3, so $U = 850 \times 3.35 \times 10^{-8} \times 9.81 = 2.79 \times 10^{-4}$ N $\approx 2.8 \times 10^{-4}$ N.
(c) At the terminal speed, drag $= W - U = 2.56 \times 10^{-3} - 2.8 \times 10^{-4} = 2.28 \times 10^{-3}$ N. So $v = F/(6\pi\eta r) = 2.28 \times 10^{-3}/(6\pi \times 0.20 \times 2.0 \times 10^{-3}) = 0.30$ m s^{-1}.



Why the spread-eagle pose? Spreading out gives the largest area, so the most drag. The bigger the drag, the sooner drag balances weight – and the lower the steady terminal velocity they fall at

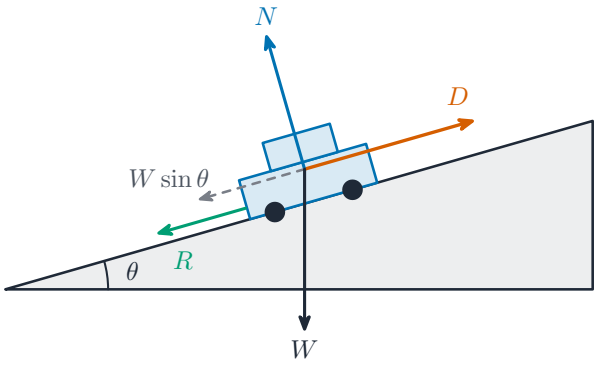
■ Energy during a terminal-velocity fall

At terminal velocity, a **parachutist** has constant kinetic energy. But the **gravitational potential energy** keeps falling as they go down. Where does it go? Almost all of it turns into **thermal energy** of the air around them. It does **not** become kinetic energy of the parachutist – that stays constant.

■ Cyclist or car at constant speed

A vehicle at constant speed on a flat road has zero resultant force. The forward **driving force** is equal and opposite to the total resistive force (friction, air resistance, and rolling resistance). At higher speed the drag is larger, so the driving force must be larger too – and so the **power** must be larger.

On a slope the weight has a component along the road. Going up at constant speed, the driving force must balance the resistive force **and** the part of the weight that pulls down the slope, $W \sin\theta$; going down, that same component helps the motion.



A car climbing a slope at constant speed: along the slope the driving force balances the resistive force plus the weight's component $W \sin\theta$

Worked example. A car of mass 1500 kg climbs a road inclined at 12° to the horizontal at a constant 30 m s^{-1}. The total resistive force is 1600 N. Find the driving force and the useful output power of the engine.

Constant speed means zero resultant force along the slope: $D = 1600 + 1500 \times 9.81 \times \sin 12° = 1600 + 3060 = 4660$ N ≈ 4700 N. Power is force times velocity (topic 5): $P = Dv = 4660 \times 30 = 1.4 \times 10^5$ W.

Conservation of linear momentum



In a crash, a large force acts over a very short time to change momentum.

The principle

For a system with no resultant external force, the total momentum stays constant. This is **conservation of momentum** . Stated for the two marks the examiner gives: **the total momentum of a system of objects remains constant provided no resultant external force acts on the system.** Both halves are needed: "total momentum is constant" alone scores one.

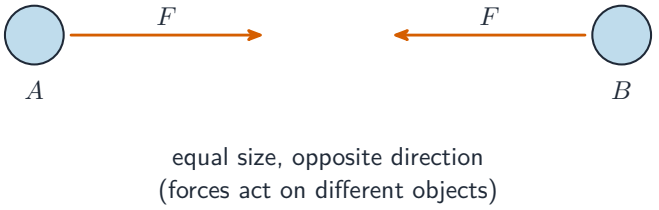
It always holds when there is no outside resultant force – in collisions, **explosions** , and **recoil** . In two dimensions, momentum is conserved along each direction on its own.

> **Worked example.** A 2.0 kg trolley and a 3.0 kg trolley are held together against a compressed spring, then released from rest. The 2.0 kg trolley flies off at 6.0 m s^{-1}. Find the speed of the other trolley.

The total momentum stays zero (it started at rest), so

$$0 = (2.0)(6.0) + (3.0)(-v) \quad \Rightarrow \quad v = \frac{12}{3.0} = 4.0 \text{ m s}^{-1}$$

in the opposite direction.



Newton's third law in an isolated two-particle system: equal and opposite forces

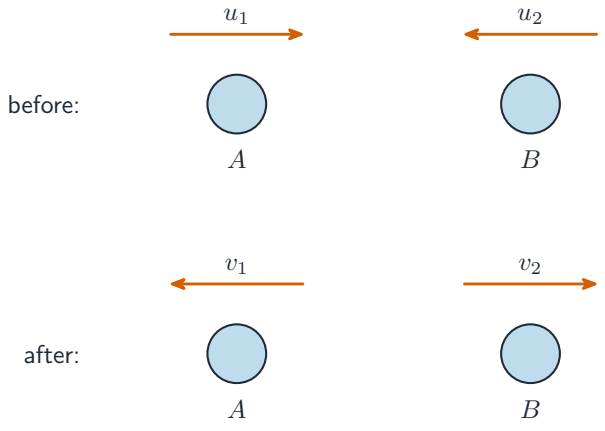
Elastic and inelastic collisions

In **every** collision, momentum is conserved (if there is no outside resultant force).

An **elastic collision** is one where the total kinetic energy is **also** conserved. A quick test: in an elastic collision, the **relative speed** of approach equals the relative speed of separation.

In an **inelastic collision** , momentum is conserved but kinetic energy goes down – some becomes thermal, sound, or **deformation** energy. If the two objects stick together, the collision is perfectly inelastic.

Solving collision problems (one dimension)



Head-on collision: velocities before and after

For two objects with masses m_1, m_2 and starting velocities u_1, u_2 that hit **head-on** , write

$$m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2.$$

Use **signed** velocities (positive in one chosen direction). If the collision is elastic, add the relative-speed equation

$$u_1 - u_2 = -(v_1 - v_2),$$

or, the same thing, $\frac{1}{2}m_1u_1^2 + \frac{1}{2}m_2u_2^2 = \frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2$. That gives two equations for two unknowns.

> **Worked example.** A 1500 kg car moving at 12 m s^{-1} runs into a stationary 1000 kg car and they lock together. Find their common velocity just after the collision.

Momentum is conserved (the cars stick, so $v_1 = v_2 = v$):

$$1500 \times 12 + 1000 \times 0 = (1500 + 1000)\,v \quad \Rightarrow \quad v = \frac{18\,000}{2500} = 7.2 \text{ m s}^{-1}.$$

Questions often go on to ask what fraction of the kinetic energy is transferred to other forms. Before: $E_\text{k} = \frac{1}{2} \times 1500 \times 12^2 = 1.08 \times 10^5$ J. After: $\frac{1}{2} \times 2500 \times 7.2^2 = 6.48 \times 10^4$ J. So 4.3×10^4 J, which is 40% of the original kinetic energy, becomes thermal energy, sound and deformation of the cars. Momentum is conserved; kinetic energy is not.

> **Worked example.** A stationary nucleus of mass $222\,u$ decays by emitting an **alpha particle** α of mass $4\,u$ at 1.6×10^7 m s^{-1}. Find the speed of the remaining nucleus.

Before the decay the total momentum is zero, so afterwards the two momenta are equal and opposite: $218\,u \times v = 4\,u \times 1.6 \times 10^7$, giving $v = 2.9 \times 10^5$ m s^{-1} in the opposite

direction to the alpha particle. The unit u cancels, so its value is never needed. In a decay the momentum, the charge (proton number), the nucleon number and the total mass–energy are all conserved (topic 11).

A useful result for a head-on elastic collision of mass m with a **stationary** mass M:

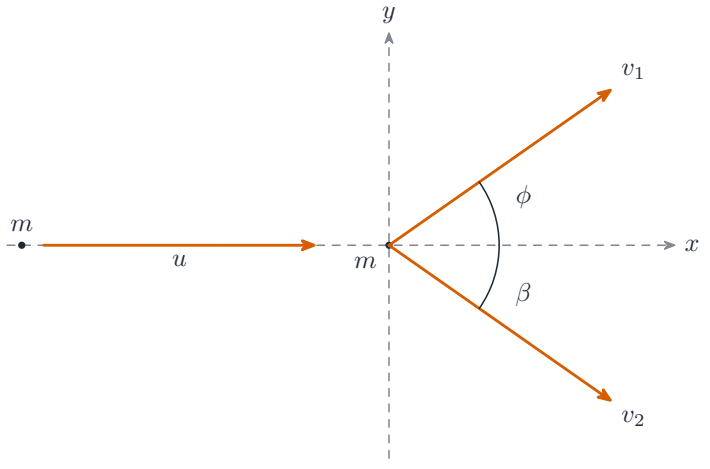
$$v_m = \frac{m - M}{m + M}u, \qquad v_M = \frac{2m}{m + M}u.$$

Collisions in two dimensions

If the objects move in two dimensions, split the velocities into **perpendicular components** and apply conservation of momentum along each direction on its own. For a collision where the objects hit at an angle, choose one axis along the first object's motion and one across it. The total momentum is conserved along each axis.

> **Worked example.** On a frictionless surface, ball A has momentum 4.0 N s due east and ball B has momentum 3.0 N s due north. They collide and stick together. Find the momentum of the combined object.

Momentum is conserved along each axis: 4.0 N s east and 3.0 N s north after the collision, exactly as before. The total momentum is the vector sum, $\sqrt{4.0^2 + 3.0^2} = 5.0$ N s at $\tan^{-1}(3.0/4.0) = 37°$ north of east; the combined object moves that way at 5.0 N s divided by the total mass. A multiple-choice diagram of two momentum arrows is asking exactly this: add the arrows tip to tail.



A glancing collision resolved along two perpendicular axes

■ Rocket / pushing out mass

A rocket pushes out gas at velocity u (relative to itself) at a **mass-flow rate** $\dot{m}$ (kg per second). It feels a thrust

$$F = \dot{m} \cdot u,$$

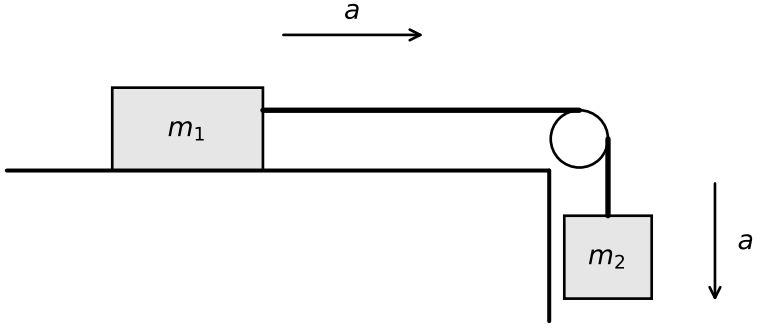
which comes from $F = \Delta p/\Delta t$. The momentum given to the gas each second equals the thrust on the rocket (Newton's third law: the rocket pushes the gas one way, the gas pushes the rocket the other way). An engine ejecting 90 kg of gas per second at 190 m s^{-1} produces a thrust of $90 \times 190 = 1.7 \times 10^4$ N.

> **Worked example.** A rocket of weight W leaves the ground with an initial acceleration a. What thrust does its engine produce?

Take upwards as positive. The resultant force is thrust minus weight, so $T - W = ma$. With $m = W/g$, $T = W + Wa/g = W(1 + a/g)$. A rocket of weight 2.0×10^6 N accelerating at 4.0 m s^{-2} needs a thrust of $2.0 \times 10^6 \times (1 + 4.0/9.81) = 2.8 \times 10^6$ N. The thrust must exceed the weight before the rocket moves at all.

■ Connected bodies moving along different lines

Draw a separate force diagram for each body. A light string over a smooth pulley has the same tension on both sides; a taut inextensible string gives the bodies the same acceleration magnitude. Choose a positive direction along each body's motion.



> **Worked example.** A 4.0 kg block on a smooth table is joined to a hanging 1.0 kg mass. For the block, $T = m_1 a$; for the hanging mass, $m_2 g - T = m_2 a$. Adding cancels the internal tension: $a = m_2 g/(m_1 + m_2) = 1.0 \times 9.81/(4.0 + 1.0) = 1.96$ m s^{-2}. Then $T = m_1 a = 4.0 \times 1.96 = 7.85$ N. Normal force and weight balance on the block, but weight and tension do not balance on the accelerating hanging mass.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
mass	the property of an object that resists a change in its motion
linear momentum	the product of an object's mass and its velocity
force	the rate of change of momentum
Newton's first law	an object remains at rest or moves at constant velocity unless acted on by a resultant force
Newton's second law	the resultant force on an object is proportional to the rate of change of its momentum, and acts in the direction of the change
Newton's third law	when two bodies interact, the force on one is equal in magnitude and opposite in direction to the force on the other
weight	the force on an object due to a gravitational field; equal to the product of its mass and the acceleration of free fall
principle of conservation of momentum	the total momentum of a system of objects remains constant provided no resultant external force acts on the system
elastic collision	a collision in which total kinetic energy is conserved (and the relative speed of approach equals the relative speed of separation)
inelastic collision	a collision in which momentum is conserved but some kinetic energy is transferred to other forms
terminal velocity	the constant velocity reached when the resistive force on a moving object equals the force driving it, so the resultant force is zero

Exam tips

- **Newton's second law is** $F = \Delta p/\Delta t$ (rate of change of momentum); $F = ma$ is the special case for constant mass.
- Identify **third-law pairs** correctly: the same type of force, acting on **two different bodies** – not the balanced forces on one body.
- **Momentum is conserved** in every collision; kinetic energy is conserved only in an **elastic** collision.

- For **terminal velocity**, explain that drag rises with speed until drag = weight, so the acceleration becomes zero.

Common mistakes

- Writing "the forces are balanced" or "the forces cancel out". Say "the resultant force on the object is zero".
- Adding both speeds in a head-on collision because momentum is "total". Choose a positive direction, give the opposing velocity a minus sign, then add.
- Calculating kinetic energy from a component of the velocity. Kinetic energy uses the full speed of each body.
- Calling the weight and the normal contact force on one object a third-law pair. A pair acts on two different objects and is one type of force.
- Stating conservation of momentum without the condition. It holds only when no resultant external force acts on the system.

Forces, density and pressure

A-Level Physics

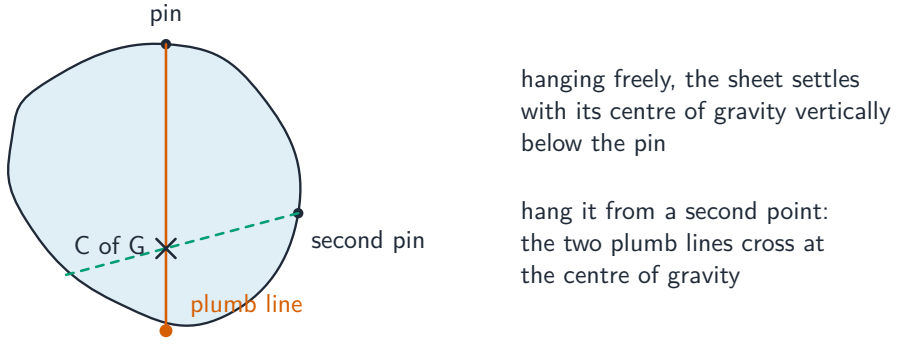
Turning effects of forces

Centre of gravity

The **weight** of a large object can be treated as acting at one single point, called the **centre of gravity** (the same as the **centre of mass** in a uniform gravitational field). For a uniform, regular shape – a rectangle, a sphere, a uniform rod – the centre of gravity is at the middle.

When you draw a **free-body diagram**, always put the weight arrow at the centre of gravity.

For an irregular shape, find the centre of gravity by experiment. Hang the object freely from a pin: in equilibrium it settles with its centre of gravity **vertically below the pin**, because only then does the weight have no moment about the pin. A **plumb line** hung from the same pin marks that vertical. Hang the object from a second point and draw the second line; the centre of gravity is where the two lines cross. The same fact answers the exam's "explain why the sheet hangs like this": its centre of gravity is directly below the support, so the weight's line of action passes through the pivot and there is no resultant torque.



Finding a centre of gravity: hung freely, an object settles with its centre of gravity vertically below the pin, so two plumb lines from two pins cross at it

For a **non-uniform** bar, the centre of gravity is not at the middle. Find it by balancing the bar on a pivot, or by taking moments: if the bar's weight W balances a known weight, the distance of its centre of gravity from the pivot is the one moment that makes the two sides equal.

Moment of a force

The **moment** of a force about a point is

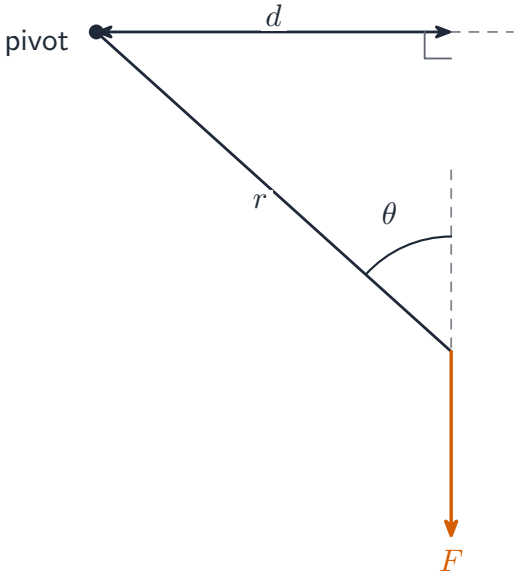
$$M = F \cdot d,$$

where F is the size of the **force** and d is the **perpendicular distance** from the point to the **line of action** of the force. Unit: N m.

If the force acts at angle θ to a **lever arm** of length r from the **pivot**, then $d = r\sin\theta$, so $M = Fr\sin\theta$. Only the part of the force **perpendicular** to the lever arm makes it turn.

> **Worked example.** A force of 50 N is applied to the end of a spanner 0.40 m long, at 30° to the spanner. Find the moment of the force about the nut.

$$M = Fr\sin\theta = 50 \times 0.40 \times \sin 30° = 50 \times 0.40 \times 0.50 = 10 \text{ N m}.$$



The moment depends on the perpendicular distance d from the pivot to the line of action of the force

A moment is either **clockwise** or **anticlockwise** about the chosen point.

Couple and torque

A **couple** is a pair of forces that are:

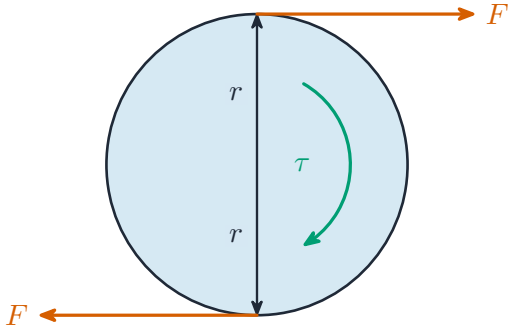
- **equal in size**,
- **opposite in direction**,
- with their lines of action a perpendicular distance apart.

A couple makes the body **turn only** – its resultant force is zero, so it gives no straight-line acceleration.

The **torque** of a couple is the turning effect it makes:

$$\tau = F \cdot d,$$

where F is the size of one force and d is the perpendicular distance between the two lines of action. Unit: N m. The torque is the same about any point – a special property of couples.



A couple: two equal and opposite forces, a distance apart, producing a torque

A common multiple-choice trap: two equal forces in the **same** direction are not a couple (they have a resultant force and cause **translation**). A couple needs equal size **and** opposite direction.

Watch the distance in the formula. For a couple of two forces F each acting a distance d from a central pivot, on opposite sides, the perpendicular distance between the two lines of action is $2d$, so the torque is $F \times 2d = 2Fd$. Taking moments about the pivot gives the same answer, $Fd + Fd$. The two-mark definition wants both parts: **the product of one of the forces and the perpendicular distance between the lines of action of the forces.**

Equilibrium of forces

Conditions for equilibrium

A body is in **equilibrium** when:

1. the **resultant force** is zero (no straight-line acceleration), AND
2. the resultant moment about any point is zero (no **angular acceleration**).

Both must hold. A body with no resultant force can still be turning; a body with no resultant moment can still be moving in a straight line.

The classic multiple-choice case: four forces, two equal pairs pointing in opposite directions but not along the same lines. The resultant force is zero, so the centre of mass does not accelerate, but the pairs form couples with a resultant torque, so the object spins faster and faster while its centre of mass moves at constant velocity. Only when both conditions hold is the body in equilibrium.

■ Principle of moments

For a body that is not turning, **the total clockwise moment about any point equals the total anticlockwise moment about the same point**. This is the **principle of moments** .

To solve a balance problem:

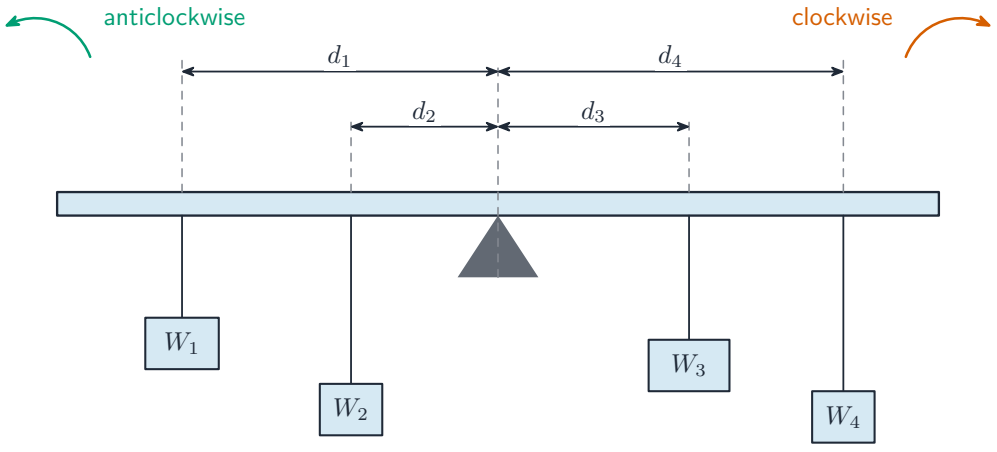
1. Choose a pivot – usually where an unknown force acts, so that force drops out (its distance is zero).
2. List every force and its perpendicular distance from the pivot.
3. Set $\sum M_{\text{clockwise}} = \sum M_{\text{anticlockwise}}$.
4. Use $\sum F = 0$ if you need a second equation.

A ruler balanced on a pivot with masses on each side is solved this way. For a heavy uniform rod, remember to include its weight acting at its centre of gravity. Stated for the marks: **for a body in equilibrium, the sum of the clockwise moments about any point equals the sum of the anticlockwise moments about that same point.**

> **Worked example.** A uniform beam of length 4.0 m is pivoted at its centre. A 40 N weight hangs 1.5 m from the pivot on one side. How far from the pivot, on the other side, must a 30 N weight hang to balance the beam?

The beam's own weight acts at the centre (the pivot), so it has no moment. Setting clockwise = anticlockwise moments:

$$40 \times 1.5 = 30 \times d \quad \Rightarrow \quad d = \frac{60}{30} = 2.0 \text{ m}.$$

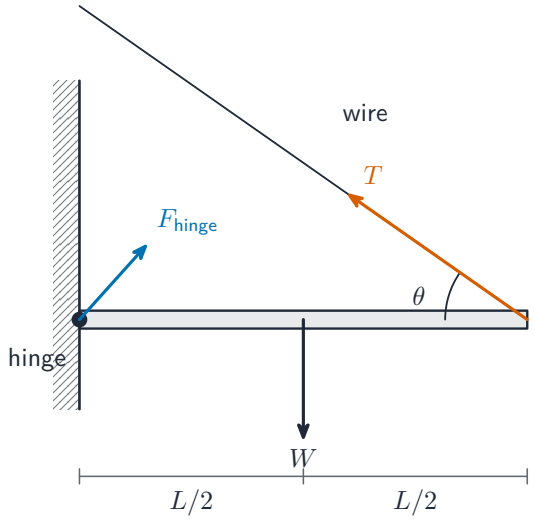


Weights on a rule balanced at a pivot – used to test the principle of moments



A tower crane is a moments problem: the counterweight's moment about the mast balances the moment of the load

Worked example. A uniform rod of length L and weight $W = 28$ N is hinged to a wall at one end and held horizontal by a wire from its other end. The wire makes $40°$ with the rod. Find the tension T in the wire.



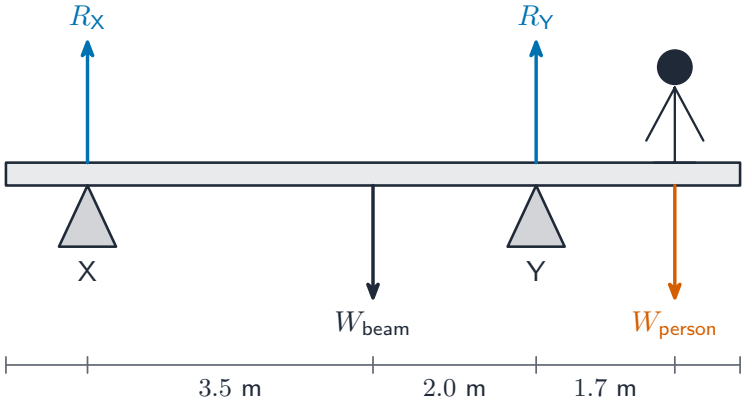
A hinged rod held by a wire: taking moments about the hinge removes the unknown hinge force

Take moments about the hinge, so the unknown hinge force drops out. The weight acts at the centre, a distance $L/2$ from the hinge: its moment is $W \times L/2$, clockwise. The tension acts at the end, but not at right angles to the rod, so its moment is $T \sin 40^\circ \times L$, anticlockwise (only the component of T perpendicular to the rod turns it). Equate them:

$$T \sin 40^\circ \times L = W \times \frac{L}{2} \quad \Rightarrow \quad T = \frac{28}{2 \sin 40^\circ} = 22 \text{ N}.$$

The length L cancels, which is why the question need not give it. If a load hangs from the rod as well, add its moment on the clockwise side. This is the shape of almost every moments question: choose the pivot where the unknown force acts, use the perpendicular component of any angled force, and let unknown lengths cancel.

> **Worked example.** A uniform beam of weight 200 N and length 9.0 m rests on two supports, X at 1.0 m from the left end and Y at 6.5 m. A person of weight 700 N stands 1.7 m from Y, beyond it. Find the force from each support.



A beam on two supports: moments about one support give the other support's force; the sum of forces then gives the first

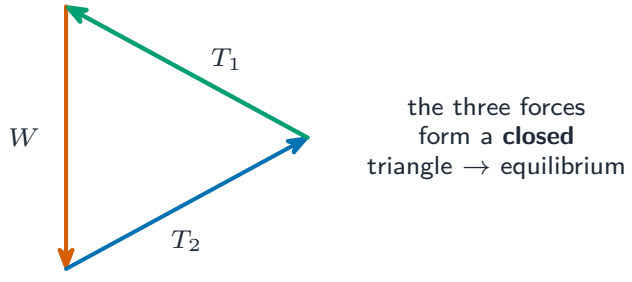
Two unknown forces, R_X and R_Y, so take moments about X to remove R_X. The beam's weight acts at its centre, 3.5 m from X; Y is 5.5 m from X; the person is 7.2 m from X:

$$R_Y \times 5.5 = 200 \times 3.5 + 700 \times 7.2 \quad \Rightarrow \quad R_Y = \frac{700 + 5040}{5.5} = 1040 \text{ N}.$$

Then the resultant force is zero: $R_X + R_Y = 200 + 700$, so $R_X = 900 - 1040 = -140$ N. A negative answer means the beam would lift off X: with the person that far past Y, X must hold the beam down, not up. Questions often ask where a person can stand before the beam tips; that is the position at which R_X becomes zero.

Vector triangle

Three forces in the same plane that are in equilibrium can be drawn as a **closed vector triangle** – drawn tip-to-tail, the three arrows come back to the start. This is a drawing method instead of splitting into **components** .



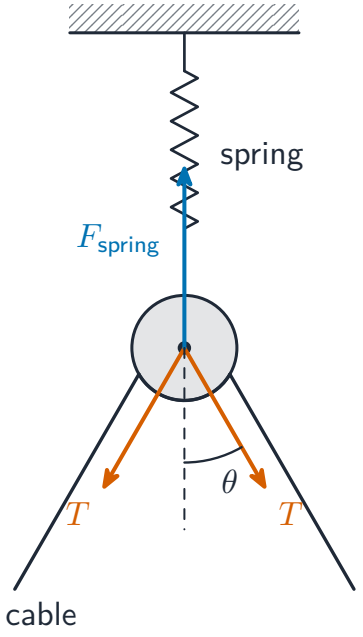
Three forces in equilibrium form a closed vector triangle

Use the **sine rule** or the **cosine rule** on the triangle to find unknown sizes or directions, or draw the triangle to scale on graph paper.

You can also split each force into **horizontal** and **vertical** components and set $\sum F_x = 0$ and $\sum F_y = 0$.

To draw the triangle for the marks: draw the weight first (vertical, to scale), then add the other two forces tip to tail in their real directions, so the third arrow closes the triangle; label every side with its force and mark the angles. For a block resting on a slope the three forces are the weight (vertical), the normal contact force (perpendicular to the slope) and friction (along the slope), so the triangle is right-angled with the weight as the hypotenuse. For a picture hanging from a string over a pin, the two equal tensions make an isosceles triangle with the weight. The test of any answer is that the triangle closes; if it does not, the object is not in equilibrium.

> **Worked example.** A pulley of negligible weight is held by a spring. A single cable passes under the pulley, and each side of the cable makes 30° with the vertical. The tension in the cable is $T = 60$ N. Find the force from the spring and the extension of the spring, given its spring constant is 2000 N m^{-1}.



A pulley held by a spring: the two tensions in the one cable are equal, and the spring force balances their vertical components

Because it is one continuous cable, the tension is the same on both sides. The two tensions pull down and outwards; their horizontal components cancel and their vertical components add, so the spring must pull up with $F = 2T\cos 30° = 2 \times 60 \times 0.866 = 104$ N. From Hooke's law (topic 6), $x = F/k = 104/2000 = 0.052$ m. If the angle to the vertical grows, $\cos\theta$ falls, so the spring force and extension fall: the cable is pulling more sideways and less upwards.

Density



An iceberg floats with most of its volume hidden: ice is slightly less dense than water.

Density is the mass per unit volume:

$$\rho = \frac{m}{V}.$$

Unit: kg m^{-3} (or g cm^{-3}; 1 g cm^{-3} = 1000 kg m^{-3}). Density is a **scalar** .

Some useful densities to know:

- water: 1000 kg m^{-3}
- air at room conditions: $\sim$ 1.2 kg m^{-3}
- iron / steel: $\sim$ 7800 kg m^{-3}

Measuring density is a practical question in disguise. For a cuboid, measure the mass on a balance and the three side lengths with a rule or calipers, then $\rho = m/(xyz)$. Because density is a product and quotient, the percentage uncertainty in ρ is the **sum** of the percentage uncertainties in m, x, y and z (topic 1). A zero error on the balance, or a rule whose end is worn, is a systematic error; repeating the readings does not remove it, but subtracting the zero reading does.

> **Worked example.** A block has mass (120 ± 1) g and sides (4.0 ± 0.1) cm, (3.0 ± 0.1) cm and (2.0 ± 0.1) cm. Find its density and the percentage uncertainty.

$\rho = 120/(4.0 \times 3.0 \times 2.0) = 5.0$ g cm^{-3} = 5000 kg m^{-3}. The percentage uncertainties are 0.8%, 2.5%, 3.3% and 5.0%, so the density is uncertain by 12%, about $\pm$600 kg m^{-3}. The shortest side contributes most, which is why a question asks which measurement to improve first.

Pressure

Pressure is the force per unit area, where the force acts at right angles to the surface:

$$p = \frac{F}{A}.$$

Unit: Pa = N m^{-2}. Pressure is a scalar.

The force is the one pressing at right angles to the surface, and the area is the area of contact. A tree trunk of weight 9.0 kN standing on a post of diameter 0.28 m presses with $p = F/A = 9000/(\pi \times 0.14^2) = 1.5 \times 10^5$ Pa; the same weight on a wider post gives a smaller pressure. Convert diameters to radii, and centimetres to metres, before squaring.



A precision aneroid barometer measures atmospheric pressure

■ Hydrostatic pressure

Take a column of **fluid** with density ρ, **cross-sectional area** A and height Δh. Its weight is

$$W = mg = (\rho \cdot A \cdot \Delta h) \cdot g.$$

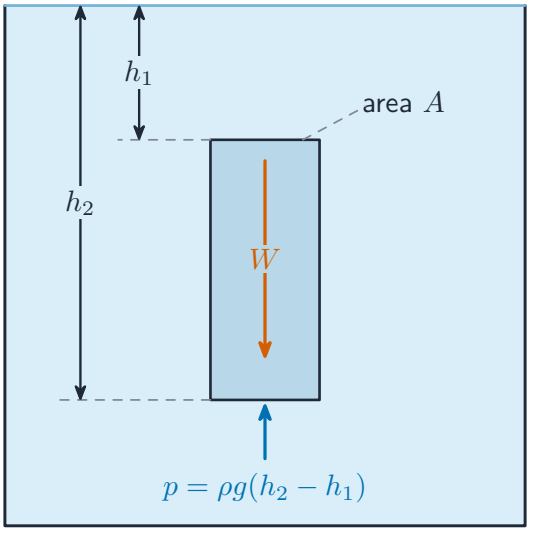
This weight presses down on the area A at the bottom, so the extra pressure at the bottom compared with the top is

$$\Delta p = \frac{W}{A} = \rho g \Delta h.$$

This is the **hydrostatic pressure** equation. It depends only on the density of the fluid and the **depth** – the shape of the container does not matter.

> **Worked example.** Find the extra pressure due to the water at the bottom of a swimming pool 2.5 m deep. (Water density 1000 kg m^{-3}, $g = 9.81$ m s^{-2}.)

$$\Delta p = \rho g \Delta h = 1000 \times 9.81 \times 2.5 \approx 2.5 \times 10^4 \text{ Pa}.$$

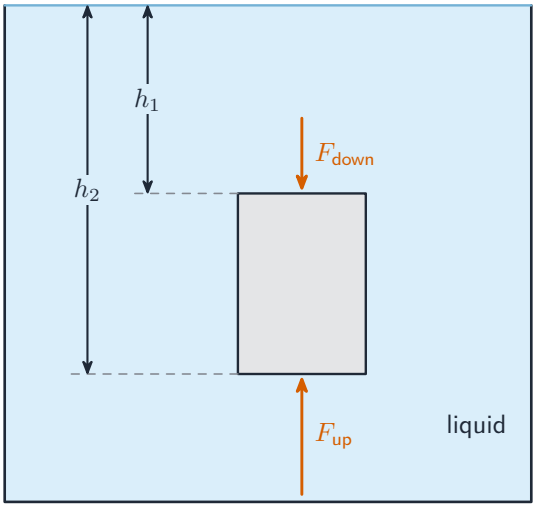


A column of liquid of area A: its weight sets the extra pressure at the depth below

For a submarine at depth h below the surface, the pressure from the water is $\rho_{\text{seawater}}\, g\, h$. For the total pressure, add the **atmospheric pressure** at the surface (about 1.0×10^5 Pa).

Upthrust and Archimedes' principle

When an object is **submerged** in a fluid, the pressure at the bottom of the object is greater than the pressure at the top (by $\rho g \Delta h$, where Δh is the object's height). This difference gives a net upward force called the **upthrust** .



Upthrust arises because the pressure on the bottom of the object is greater than on the top

For an object of volume V (the volume of fluid **displaced**), the upthrust is

$$F_{\text{upthrust}} = \rho_{\text{fluid}}\, g\, V.$$

This is **Archimedes' principle** : the upthrust on a body in a fluid equals the weight of the fluid it pushes aside.

> **Worked example.** A metal block of volume 2.0×10^{-3} m^3 is fully submerged in water. Find the upthrust on it. (Water density 1000 kg m^{-3}, $g = 9.81$ m s^{-2}.)

$$F_{\text{upthrust}} = \rho_{\text{fluid}}\, g\, V = 1000 \times 9.81 \times 2.0 \times 10^{-3} \approx 20 \text{ N}.$$

For a fully submerged object, V is its full volume. For a **floating** object, V is only the volume below the surface – the object floats when the upthrust on the part below the surface equals its weight.

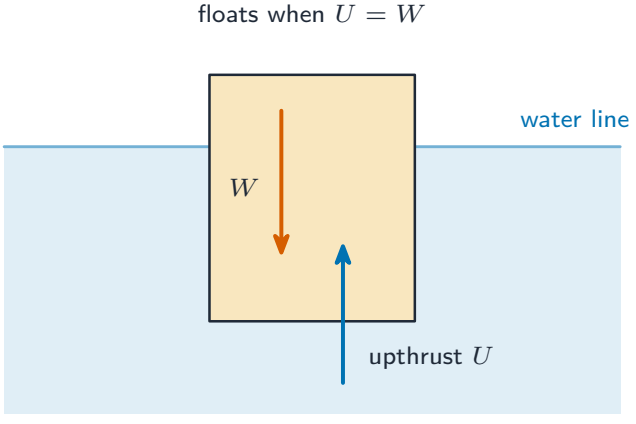
Whether an object floats at all depends on densities: for a fully submerged object the ratio of upthrust to weight is $\rho_{\text{fluid}} V g / \rho_{\text{object}} V g = \rho_{\text{fluid}} / \rho_{\text{object}}$, so it floats if it is less dense than the fluid and sinks if it is denser. Halving the object's density doubles that ratio; changing its volume or the depth changes nothing.

> **Worked example.** A cylinder of mass 11 kg and diameter 0.78 m floats upright in water of density 990 kg m^{-3}. Find the depth y of its base below the surface.

Floating means upthrust = weight. The submerged volume is the base area times the depth, Ay, with $A = \pi \times 0.39^2 = 0.478$ m^2:

$$\rho g A y = mg \quad \Rightarrow \quad y = \frac{m}{\rho A} = \frac{11}{990 \times 0.478} = 0.023 \text{ m}.$$

Each extra newton of load on the cylinder needs an extra $1/(\rho g A)$ of depth, so the depth rises in a straight line with the added weight, wherever on the cylinder it is placed.



A floating object sinks until the upthrust on the submerged part equals its weight

Force balance with upthrust

A block held under water by a string tied to the bottom of the container is in equilibrium under three vertical forces: weight (down), **tension** (down), upthrust (up). Set $F_{\text{upthrust}} = W + T$ to find the tension.

A submerged block hanging from a **newton meter** reads less than its weight in air, because of the upthrust: reading $= W - F_{\text{upthrust}}$.

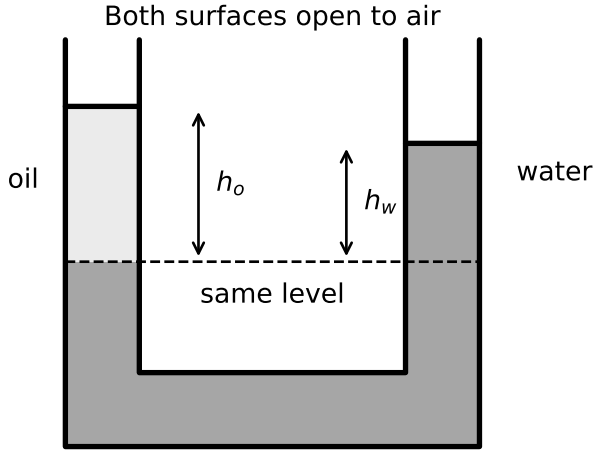
> **Worked example.** A cylinder of weight 25.0 N hangs from a newton meter fully submerged in water, and the meter reads 10.0 N. Find the volume of the cylinder.

The upthrust is the missing 15.0 N, and upthrust $= \rho g V$, so $V = 15.0/(1000 \times 9.81) = 1.53 \times 10^{-3}$ m^3. The same method finds the density of the cylinder: $25.0/(9.81 \times 1.53 \times 10^{-3}) = 1670$ kg m^{-3}.

The upthrust depends on the fluid density and the displaced volume, not on the object's material or depth (for an **incompressible** fluid). On a planet with smaller g, the upthrust is smaller in the same ratio as the weight, so a floating object still floats with the same fraction below the surface.

connected liquids

In a stationary connected liquid, pressures at the **same horizontal level in the same liquid** are equal. Start at a free surface, add $\rho g h$ when moving down, and subtract it when moving up. Both open surfaces have atmospheric pressure, which cancels.



> **Worked example.** Oil of density 800 kg/m^3 has a 0.10 m column above the oil–water boundary in an open U-tube. The water surface is h_w above that level. $p_{\text{atm}} + \rho_o g h_o = p_{\text{atm}} + \rho_w g h_w$, hence $h_w = \rho_o h_o / \rho_w = 800 \times 0.10/1000 = 0.080$ m. Compare heights from the same level, not from the tube's bottom.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
centre of gravity	the point at which the whole weight of an object may be considered to act
moment of a force	the product of the force and the perpendicular distance from the point to the line of action of the force
couple	a pair of equal and opposite forces whose lines of action do not coincide, which produces rotation only
torque of a couple	the product of one of the forces and the perpendicular distance between the lines of action of the forces
principle of moments	for a body in equilibrium, the sum of the clockwise moments about any point equals the sum of the anticlockwise moments about that point
equilibrium	the state of a body on which the resultant force and the resultant torque are both zero
density	mass per unit volume
pressure	force per unit area, where the force acts at right angles to the surface
upthrust	the upward force on a body in a fluid caused by the pressure being greater on its lower surface than on its upper surface
Archimedes' principle	the upthrust on a body in a fluid is equal to the weight of the fluid displaced by the body

Exam tips

- Take **moments** about a chosen pivot; for equilibrium, total clockwise = total anticlockwise moments **and** the resultant force is zero.
- A body in equilibrium under three forces gives a **closed triangle** of forces.
- Fluid pressure = ρgh; **upthrust = weight of fluid displaced** (Archimedes).
- Distinguish mass, weight and density, and give the base unit each time.

Common mistakes

- Resolving with $\cos\theta$ where the geometry needs $\sin\theta$. Draw the triangle and check which side the angle is next to before choosing.
- Substituting the weight of the object where the question is about the upthrust on it. Upthrust is the weight of the fluid displaced, $\rho_{\text{fluid}} V g$.
- Leaving out the weight of a uniform beam. It acts at the centre, and it has a moment about any pivot that is not at the centre.
- Using the distance along the rod for an angled force. The moment needs the perpendicular distance, or the perpendicular component of the force.
- Taking the torque of a couple as Fd with d measured from the pivot. The distance is between the two lines of action, so a symmetric couple gives $2Fd$.

Work, energy and power

A-Level Physics



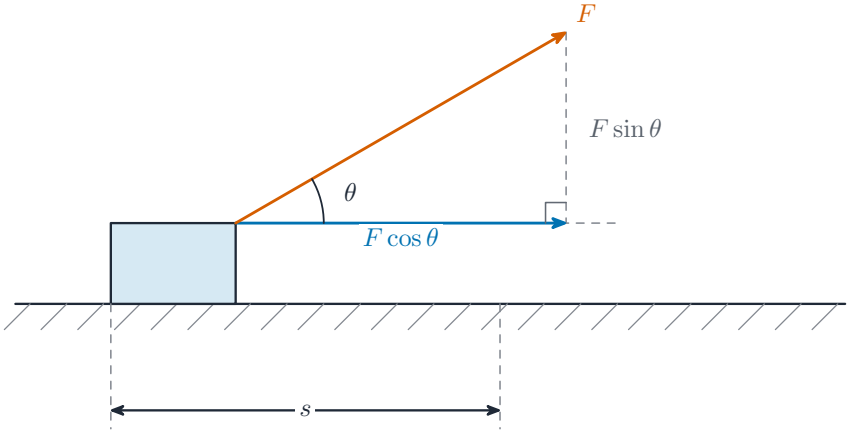
Wind turbines transfer the kinetic energy of the wind into electrical energy.

Work done by a force

Work is done when a force moves its point of contact along the line of the force. For the one-mark definition write: **work done is the product of the force and the distance moved in the direction of the force.** The work done by a constant force F that causes a **displacement** s is

$$W = F \cdot s \cdot \cos\theta,$$

where θ is the angle between the force and the displacement. Only the **component** of the force along the displacement does work.



Only the component of the force along the displacement ($F\cos\theta$) does work

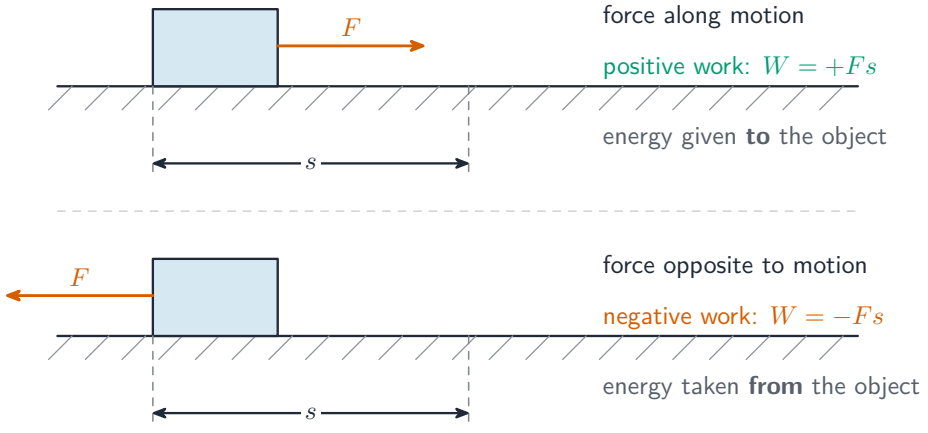
Worked example. A child pulls a sledge 5.0 m across the snow with a rope, using a force of 20 N at 60° to the ground. Find the work done by the rope.

$$W = Fs\cos\theta = 20 \times 5.0 \times \cos 60° = 20 \times 5.0 \times 0.50 = 50 \text{ J}.$$

Unit: J = N m. Work is a **scalar** .

Special cases:

- force in the same direction as the motion ($\theta = 0$): $W = Fs$, positive work, **energy** given **to** the object.
- force at right angles to the motion ($\theta = 90°$): $W = 0$. The **normal contact force** on a car on a flat road does no work.
- force opposite to the motion ($\theta = 180°$): $W = -Fs$, negative work, energy taken **from** the object (for example **friction**).

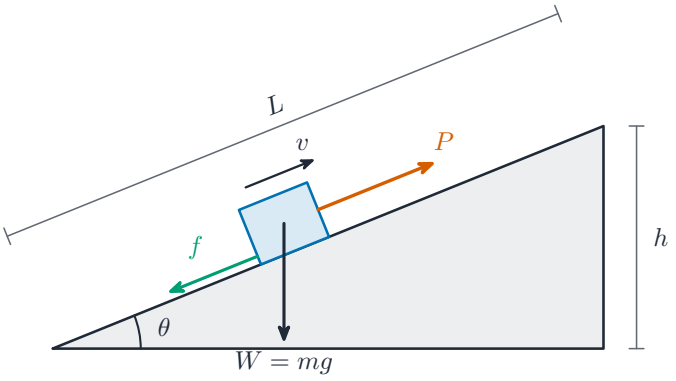


Positive work ($W = +Fs$): force along the motion (top). Negative work ($W = -Fs$): force opposite to the motion, e.g. friction (bottom)

For an object moving up a slope at angle α to the **horizontal** , the work done against gravity in rising a height h is mgh, while the work done by a horizontal push over the slope length L uses $\cos\alpha$.

Two multiple-choice tests of the definition: no work is done on an object that slides at constant velocity along a frictionless surface (no force along the motion), or on one that is simply held still (no distance moved); work is done when a force lifts it. And when a box is pushed at constant velocity across a rough floor, the work done by the push is $F \times d$ for the whole distance, however the path is split into stages.

> **Worked example.** A block of mass 4.9 kg is pushed at constant velocity up a rough slope of length 8.0 m that rises 1.5 m, by a force of 12 N acting along the slope. Find the work done against friction.



A box pushed up a rough slope at constant velocity: the work done by the push pays for the gain in GPE, which needs only the vertical rise h, and for the work against friction along the whole length L

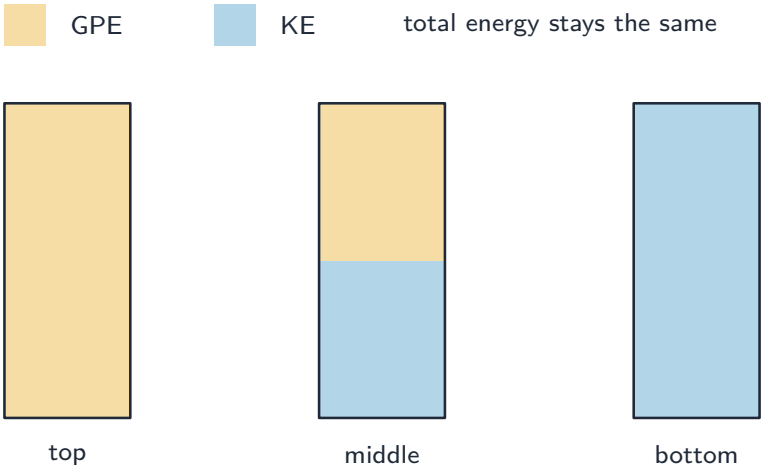
Constant velocity means no change in kinetic energy, so the work done by the push is shared between the gain in gravitational potential energy and the work done against friction. Work by the push: $12 \times 8.0 = 96$ J. Gain in GPE, using the vertical rise: $4.9 \times 9.81 \times 1.5 = 72$ J. So the work done against friction is $96 - 72 = 24$ J, and the friction force is $24/8.0 = 3.0$ N. Using the slope length in mgh is the commonest error on this question.

■ Conservation of energy

Energy is **never made or destroyed** – it only changes from one form to another, or moves from one object to another. In a **closed system**, the total energy stays constant. This is **conservation of energy**, stated for the marks as: **energy cannot be created or destroyed; it can only be transferred from one form to another** (so the total energy of a closed system is constant).

When you write an energy equation, list every form the energy starts as and ends as. Common forms in this syllabus: kinetic, gravitational potential, **elastic potential energy**, **electrical**, thermal, sound, **chemical energy**.

A ball rolling down a **frictionless ramp** turns **gravitational potential energy** into **kinetic energy**: $mgh = \frac{1}{2}mv^2$, so $v = \sqrt{2gh}$. With friction, some of this energy becomes **thermal energy** of the ramp and the air.



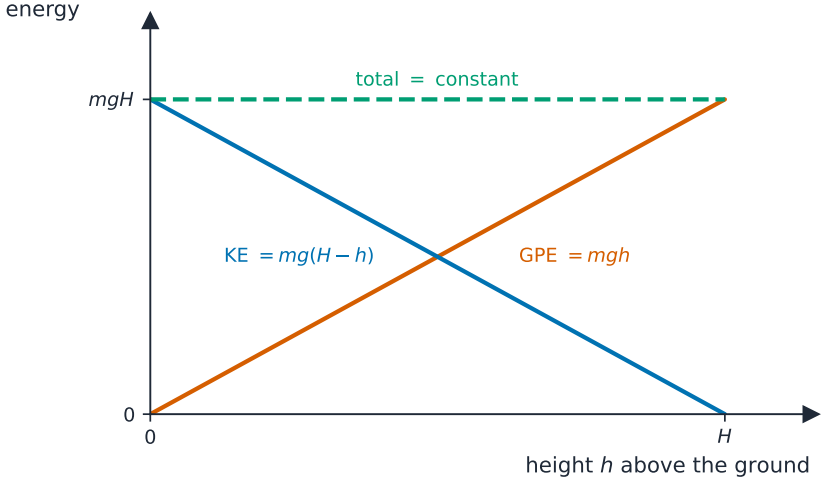
On a frictionless ramp, GPE turns into KE while the total energy stays constant

> **Worked example.** A ball is released from rest at the top of a smooth ramp 1.2 m high. Find its speed at the bottom (take $g = 9.81$ m s^{-2}).

All the gravitational potential energy becomes kinetic energy, so $v = \sqrt{2gh}$ (the mass cancels):

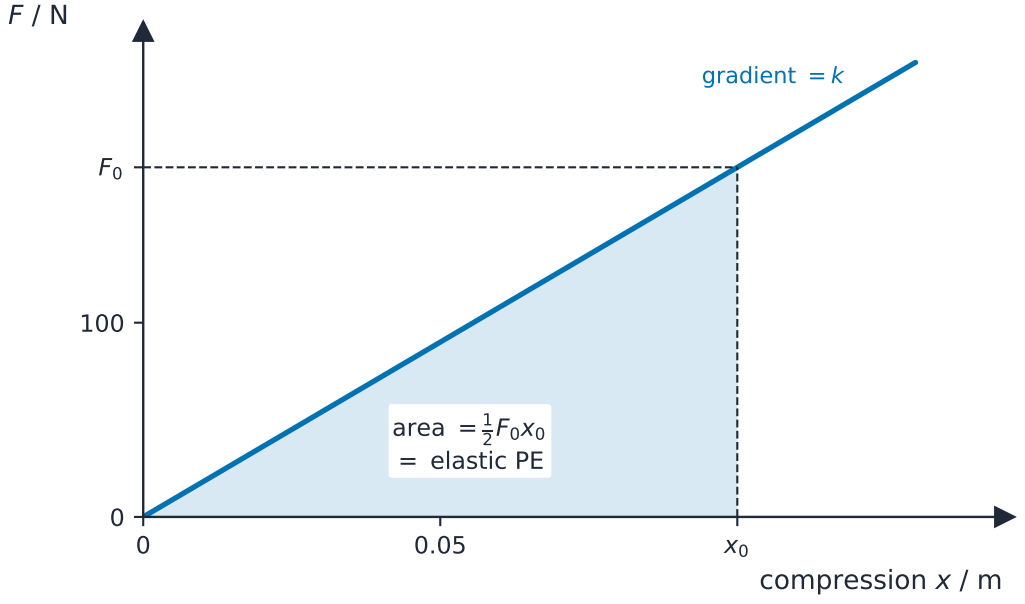
$$v = \sqrt{2 \times 9.81 \times 1.2} \approx 4.9 \text{ m s}^{-1}.$$

The same idea reads a graph. For a ball falling from height H with air resistance negligible, the GPE falls in a straight line with height (mgh), the KE rises in a straight line ($mg(H - h)$), and the two add to the constant mgH at every height.



A falling ball: GPE and KE both change in a straight line with height, and their sum is constant

The energy chain can pass through a spring. A block that hits a spring with kinetic energy 110 J while sliding a little downhill, so that its GPE falls by a further 20 J before the spring stops it, stores $110 + 20 = 130$ J of elastic potential energy at the greatest compression, if nothing is lost to friction. The spring's force–compression graph then gives the compression: the energy stored is the **area** under the line, $\frac{1}{2}F_0x_0$ for a spring that obeys Hooke's law (topic 6), so a maximum force of 2600 N means $x_0 = 2 \times 130/2600 = 0.10$ m. At that instant the resultant force on the block is the spring force minus the component of the weight down the slope, and its acceleration follows from $F = ma$ (topic 3).



The energy stored in a spring is the area under its force–compression graph: $\frac{1}{2}F_0x_0$, which equals $\frac{1}{2}kx_0^2$

Efficiency

The **efficiency** of a system is

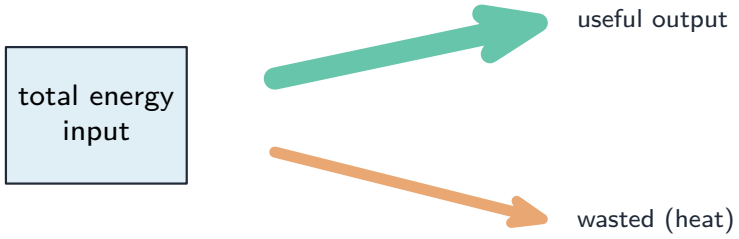
$$\text{efficiency} = \frac{\text{useful energy output}}{\text{total energy input}} \times 100\%.$$

The same idea with power:

$$\text{efficiency} = \frac{\text{useful power output}}{\text{total power input}} \times 100\%.$$

Efficiency is always less than 100% in a real system, because some input energy becomes "useless" forms – usually thermal energy. If a total energy E is supplied and an amount Q is wasted, the useful output is $E - Q$ and the efficiency is $(E - Q)/E$;

a motor rated at 1.7 kW input with efficiency 53% delivers $0.53 \times 1.7 = 0.90$ kW of useful output.



$$\text{efficiency} = \frac{\text{useful output}}{\text{total input}} \times 100\%$$

Efficiency: only part of the energy input leaves as useful output; the rest is wasted, usually as heat

For an electric motor lifting a load with efficiency η at **voltage** V and **current** I, the useful output power is $\eta V I$. From this you can find a force, a lifting speed, or a **tension** .

> **Worked example.** An electric motor lifts a 50 kg load at a steady 0.40 m s^{-1} while drawing 250 W of electrical power. Find its efficiency (take $g = 9.81$ m s^{-2}).

The useful output power is $P = mgv = 50 \times 9.81 \times 0.40 = 196$ W, so

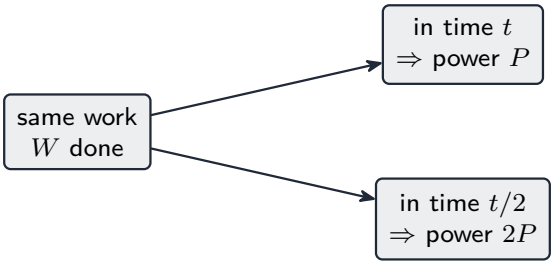
$$\text{efficiency} = \frac{196}{250} \times 100\% \approx 78\%.$$

> **Worked example.** A motor raises a block with a force of 240 N through a vertical distance of 12 m in 60 s. The input power to the motor is 900 W. Find the useful output power and the efficiency.

Work done on the block: $W = Fd = 240 \times 12 = 2880$ J. Useful output power: $P = W/t = 2880/60 = 48$ W. Efficiency: $48/900 \times 100\% = 5.3\%$. Most of the input goes into thermal energy in the motor and its cable. Turned round, a lift motor of useful output 20 kW raising 1500 kg through 20 m takes $t = mgh/P = 1500 \times 9.81 \times 20/20\,000 = 15$ s.

Power

Power is the rate of doing work, or the rate of transferring energy:



$P = \dfrac{W}{t}$ — less time, more power

The same work done in less time means more power

$$P = \frac{W}{t} = \frac{\Delta E}{\Delta t}.$$

Unit: W = J s^{-1}. Power is a scalar.

■ Power, force and velocity

For an object moving at **velocity** v with a force F along the direction of motion, in a short time Δt the displacement is $v\,\Delta t$ and the work done is $Fv\,\Delta t$. Dividing by Δt:

$$P = Fv.$$

This is one of the most useful results in mechanics. A "derive $P = Fv$" answer needs exactly those three lines: $P = W/t$, $W = Fs$ for a force along the motion, and $s/t = v$; then use it to solve problems in which a vehicle or a load moves at constant speed.

> **Worked example.** A car travels at a steady 25 m s^{-1} against a total resistive force of 600 N. Find the output power of its engine.

At constant speed the driving force equals the resistive force, so

$$P = Fv = 600 \times 25 = 15\,000 \text{ W} = 15 \text{ kW}.$$

- For a car at constant velocity v on a flat road, the engine power must balance the total resistive force: $P = F_{\text{resist}} \cdot v$. If the **drag** grows with v^2, doubling the speed multiplies that drag force by four and the power needed to balance it by **eight**: $P = Fv \propto v^3$. This assumes the stated drag is the resistive force being balanced.
- For lifting a **weight** mg straight up at constant speed v, the useful output power is $P = mg \cdot v$.
- For an aircraft hovering at a fixed height, the lift force equals the weight, and a large power is needed because air must be pushed downwards all the time.

- A crane raising 600 kg at a steady 12 m per minute lifts at $v = 0.20$ m s^{-1}, so its useful output power is $mgv = 600 \times 9.81 \times 0.20 = 1.2$ kW. Convert the speed to m s^{-1} first.
- A sailboat pushed by a constant wind force at constant velocity must also feel an equal resistive force from the water: constant velocity means zero resultant force, so the wind's power Fv is all going into work against the water.



A hydroelectric station is rated in megawatts: the energy of the stored water, divided by the time it takes to release it

Gravitational potential energy

In a **uniform** gravitational field (close to a planet's surface), the change in gravitational potential energy of **mass** m rising or falling through a height Δh is

$$\Delta E_{\mathrm{P}} = mg\Delta h.$$

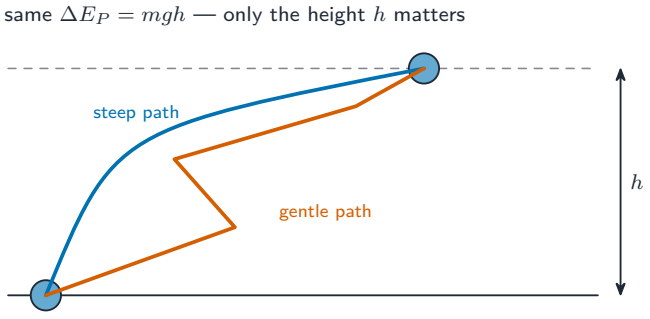
■ Where it comes from

The work done **against** gravity to raise a mass m slowly (no change in kinetic energy) through height Δh equals the gravitational potential energy gained:

- the gravitational force on the mass is mg downwards,
- the force needed to lift it slowly is mg upwards,
- the work done by this force is $W = F \cdot s = mg \cdot \Delta h$,
- this work becomes ΔE_{P}.

So $\Delta E_{\mathrm{P}} = mg\Delta h$. To use it you need m and Δh (and g). You do not need speed or time. The two-mark derivation wants the same four lines, with the condition stated:

the block is raised **at constant speed**, so the lifting force equals the weight mg, and the work done by that force over the vertical distance Δh is $mg\Delta h$, which is the gain in gravitational potential energy. A 100 g object falling 10 m loses $0.100 \times 9.81 \times 10 = 9.8$ J, about 10 J; the mass must be in kilograms.



Gravitational PE depends only on the height risen, not the path taken



A coaster train hauled up the lift hill: the work done against gravity is stored as gravitational potential energy

Kinetic energy

The kinetic energy of an object of mass m moving at speed v is

$$E_{\mathrm{K}} = \tfrac{1}{2}mv^2.$$

Where it comes from

Apply a resultant force F to a mass m that starts at rest. It speeds up evenly from 0 to v over a displacement s. From $v^2 = u^2 + 2as$ with $u = 0$,

$$s = \frac{v^2}{2a}.$$

The work done on the mass is

$$W = F \cdot s = ma \cdot \frac{v^2}{2a} = \tfrac{1}{2}mv^2.$$

All this work becomes kinetic energy, so $E_\mathrm{K} = \frac{1}{2}mv^2$. In the exam, state the assumptions as you go: the object starts from rest, the force is constant, so the acceleration is uniform, and $F = ma$ is Newton's second law.

Because $E_\mathrm{K} \propto v^2$, quadrupling the speed multiplies the kinetic energy by sixteen: an object with 1500 J at 10 m s^{-1} has 24 000 J at 40 m s^{-1}. The same square is why a stopping force is found from energy: a ball of mass 1.2 kg moving at 3.0 m s^{-1} that is stopped by a cushion over 0.020 m loses $\frac{1}{2} \times 1.2 \times 3.0^2 = 5.4$ J, so the average force on it is $5.4/0.020 = 270$ N: the work done by the cushion equals the kinetic energy lost.

Kinetic energy and momentum

Combining $p = mv$ and $E_\mathrm{K} = \frac{1}{2}mv^2$:

$$E_\mathrm{K} = \frac{p^2}{2m}.$$

This is handy when the **momentum** is given but not the velocity. For a momentum change from p_1 to p_2 at constant mass, the change in kinetic energy is $(p_2^2 - p_1^2)/(2m)$.



A pole vaulter: kinetic energy from the run-up is stored in the bent pole, then given back as height

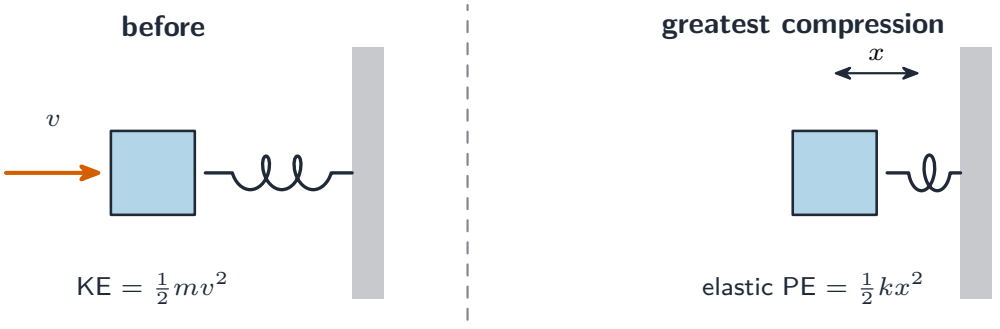
Using energy methods

A useful plan for problems that mix forces and energy:

1. **Find the start and end states.** Write the kinetic and potential energies in each.
2. **List any work done by outside forces** (friction, a push). Friction usually takes energy out; a push can add it.
3. **Conservation of energy**: $E_{\text{start}} + W_{\text{in}} = E_{\text{end}} + W_{\text{lost as heat etc.}}$.

Examples:

- A box pushed at **constant velocity** up a ramp of length L rising by h: E_{K} does not change, so the work done by the push goes into ΔE_{P} plus the work done against friction.
- A block sliding into a **spring** with kinetic energy E_{K} on a frictionless surface: at greatest **compression** x, all the kinetic energy has become elastic potential energy $\frac{1}{2}kx^2$ (where k is the **spring constant**).
- A ball dropped from height h_1 that bounces to height h_2: the ratio h_2/h_1 is the fraction of mechanical energy kept, $h_2/h_1 = (v_{\text{up}}/v_{\text{down}})^2$.
- A **projectile** thrown to the same height at different angles: the final speed is the same (only the height matters); use components to get its direction.
- A **bungee jumper**: from the platform to the point where the cord goes taut, GPE becomes KE; from there the cord stretches and takes energy as elastic potential energy, so the KE reaches its maximum where the cord's pull first equals the weight, then falls to zero at the lowest point, where GPE lost = elastic PE stored (plus any thermal energy).



A block's kinetic energy becomes elastic potential energy as it compresses the spring

■ Power when the force is at an angle

Only the component parallel to the velocity transfers energy: $P = Fv\cos\theta$, where θ is the angle between the force and velocity. It follows from $W = Fs\cos\theta$ and $P = W/t$.

> **Worked example.** A 120 N pull acts at 60° to the velocity of a trolley moving at 2.0 m/s. $P = Fv\cos\theta = 120 \times 2.0 \times \cos 60° = 120$ W. The perpendicular component does no work.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
work done by a force	the product of the force and the distance moved in the direction of the force
principle of conservation of energy	energy cannot be created or destroyed, only transferred from one form to another, so the total energy of a closed system is constant
efficiency	the ratio of useful energy (or power) output from a system to the total energy (or power) input, usually as a percentage
power	the work done per unit time, or the rate at which energy is transferred
gravitational potential energy	the energy an object has because of its position in a gravitational field
kinetic energy	the energy an object has because of its motion
elastic potential energy	the energy stored in an object that has been stretched or compressed

Exam tips

- **Work = force × distance moved in the direction of the force** – use $Fs\cos\theta$ when they are at an angle.
- Use **conservation of energy**: loss in GPE = gain in KE (+ work done against resistance).
- **Power = work / time** $= Fv$; efficiency = useful output ÷ total input.
- GPE uses the **vertical** height gained, not the distance along a slope.

Common mistakes

- Giving the change in height as the answer to a change-in-energy question. Finish the calculation: $\Delta E_{\mathrm{P}} = mg\Delta h$.
- Using the distance along a slope as the height in $mg\Delta h$, or using mass where weight is needed. Use the vertical height gained, and check whether the quantity wanted is m in kg or W in N.
- Putting the resistive force into $P = Fv$ for a car that is accelerating. F is the driving force; only at constant speed does it equal the resistive force.
- Efficiency above 100%, or efficiency as output divided by wasted. It is useful output divided by total input, and the total input is always the larger number.
- Reading a force–extension graph's gradient when the question wants the energy. Energy stored is the area under the line, not its slope.

Deformation of solids

A-Level Physics

Forces that cause deformation

When a force acts on a solid along its length, the object changes shape (**deformation**). Two cases (treated as one-dimensional here):

- a **tensile** force stretches the object – it makes an **extension** x,
- a **compressive** force squeezes the object – it makes a **compression** , treated as a negative extension.

The applied force is the **load** . The change from the natural length is the extension (or compression).

Both are measured from the natural, unstretched length, never from the loaded length: on a diagram of a spring under a box, the distance the spring has shortened is its compression. A spring's own mass is usually said to be **negligible** , so the only forces on it are the load and the tension it provides.

Hooke's law and the spring constant



A spring obeys Hooke's law: extension is proportional to the force applied.

For many materials at small extensions, **the extension is proportional to the load** – this is **Hooke's law** . The constant that links them is the **spring constant** k:

$$F = kx \qquad \Longleftrightarrow \qquad k = \frac{F}{x}.$$

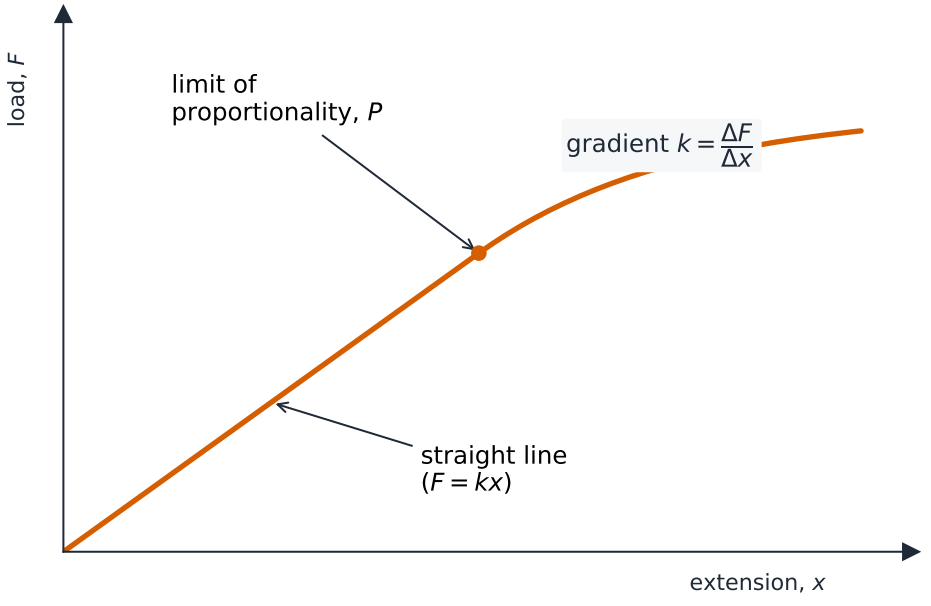
Unit of k: N m^{-1}.

> **Worked example.** A spring stretches by 4.0 cm when a 2.0 N load is hung from it. Find its spring constant.

Converting the extension to metres (4.0 cm = 0.040 m):

$$k = \frac{F}{x} = \frac{2.0}{0.040} = 50 \text{ N m}^{-1}.$$

For the one-mark "state Hooke's law", write it in words: **the extension is directly proportional to the applied force (load), provided the limit of proportionality is not exceeded.** The spring constant is the **force per unit extension**; the multiple-choice distractors are "extension per unit force" (that is $1/k$) and $\frac{1}{2}Fx$ (the energy stored).



A load–extension graph: straight up to the limit of proportionality P, then it curves

Reading a graph:

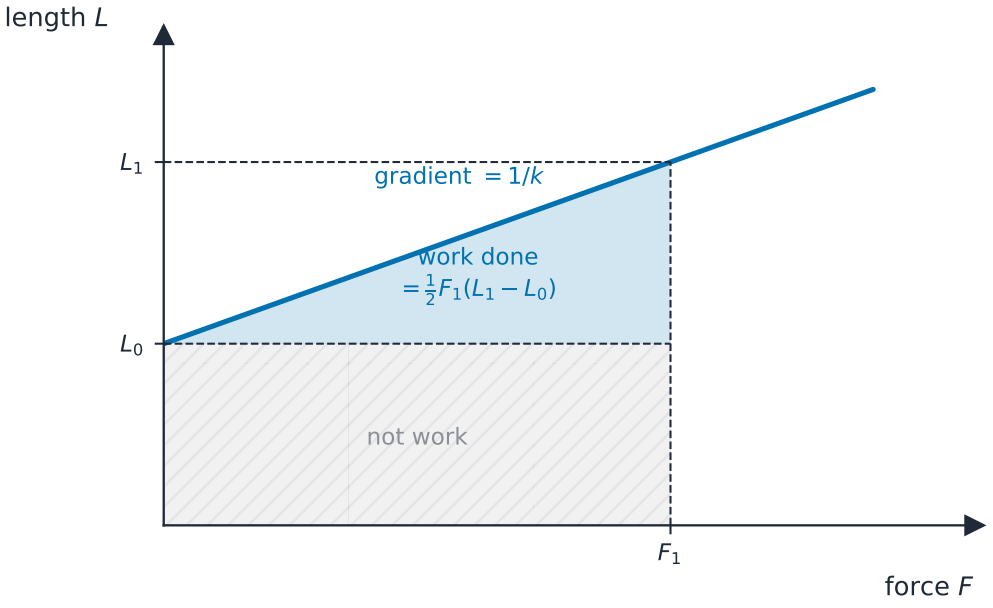
- A force–extension (F against x) graph has gradient k in the Hooke's-law region.
- An extension–force (x against F) graph has gradient $1/k$ in the Hooke's-law region.

A common trap: if a graph plots **length** L against force, you can still find the spring constant from the gradient (since $L = L_0 + F/k$, the gradient is $1/k$ – read it off carefully).

> **Worked example.** A spring has length 90 mm under a force of 0.80 N and 115 mm under 1.30 N. Find the spring constant and the unstretched length.

The extra force 0.50 N produces the extra extension 25 mm = 0.025 m, so k =

$0.50/0.025 = 20$ N m^{-1}. At 0.80 N the extension is $x = F/k = 0.80/20 = 0.040$ m, so the unstretched length is $90 - 40 = 50$ mm. On a graph of length against force the intercept on the length axis is L_0, the gradient is $1/k$, and the work done in stretching the spring is the **triangle between the line and the level** $L = L_0$, not the whole area down to the force axis: the rectangle below L_0 is the unstretched length multiplied by the force, which means nothing.



Length against force: the intercept is L_0, the gradient is $1/k$, and only the triangle above L_0 is the work done

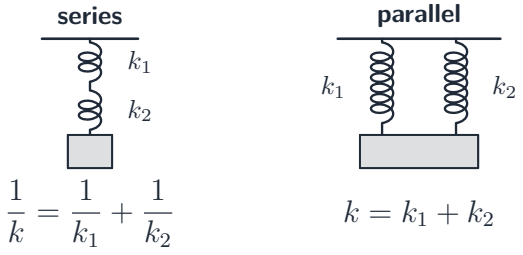
Limit of proportionality

Hooke's law only holds up to the **limit of proportionality** . Past this point the F against x line curves and is no longer straight. The material may still be **elastic** (it returns to its first length when you remove the load) a little further, then it becomes **plastic** .

Springs in series and parallel

You may need to combine spring constants:

- **Series** (one spring hangs from another): the same load passes through both, the total extension is the sum, so $\dfrac{1}{k_{\text{total}}} = \dfrac{1}{k_1} + \dfrac{1}{k_2}$.
- **Parallel** (two springs side by side holding the same load): each takes half the load (if they are identical), the extensions are equal, so $k_{\text{total}} = k_1 + k_2$.



Combining spring constants: series gives a softer spring, parallel a stiffer one

> **Worked example.** Springs of spring constant 6.0 N cm^{-1} and 4.0 N cm^{-1} are joined end to end and a load of 80 N hangs from them. Find the total extension.

Each spring carries the full 80 N, so the extensions are $80/6.0 = 13.3$ cm and $80/4.0 = 20$ cm: a total of 33 cm (the same as $k = 2.4$ N cm^{-1} from $1/k = 1/6.0 + 1/4.0$). Three identical springs give the largest combined constant when all three are in parallel ($3k$) and the smallest when all are in series ($k/3$). When a load is shared by two identical springs in parallel, each carries **half the force** and so stores a **quarter of the energy** of one spring holding the whole load; the pair together stores half as much as the single spring did. A spring constant in N cm^{-1} is 100 times smaller than the same constant in N m^{-1}; give the unit the question asks for.

Stress, strain and the Young modulus

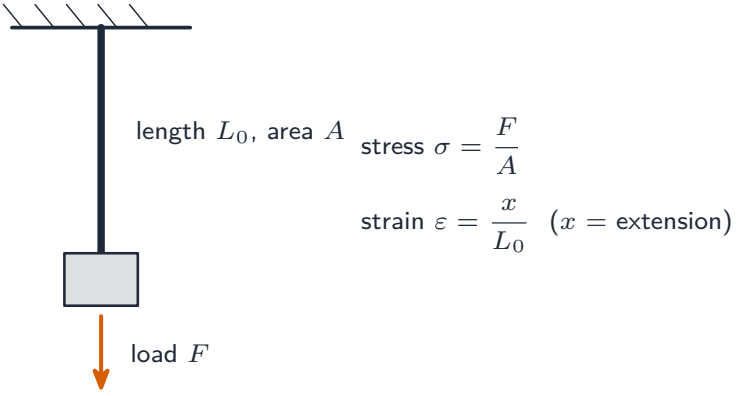
For a wire of uniform cross-section under a tensile load:

- **Stress** $\sigma = \dfrac{F}{A}$, where F is the load and A is the **cross-sectional area** . Unit: Pa.
- **Strain** $\varepsilon = \dfrac{x}{L_0}$, where x is the extension and L_0 is the original length. Strain has no unit (it is a ratio of lengths).

For the one-mark definitions: **stress is the force per unit cross-sectional area** (the force acting normally to the area); **strain is the extension per unit original length**. The unit of stress, Pa = N m^{-2}, is kg m^{-1} s^{-2} in base units. The area of a round wire comes from its diameter d: $A = \pi d^2/4 = \pi r^2$, and using d in place of r makes the area four times too big. Convert mm^2 to m^2 with 10^{-6}.

> **Worked example.** A tensile force of 18 N acts on a wire of cross-sectional area 3.2 mm^2. Find the stress.

$\sigma = F/A = 18/(3.2 \times 10^{-6}) = 5.6 \times 10^6$ Pa. The same force acts along the whole wire, so a bolt whose diameter is $2d$ at one end and d at the other has four times the stress at the narrow end ($\sigma \propto 1/d^2$), and a wire of three times the radius carries one ninth of the stress.



Stress is the load per cross-sectional area; strain is the extension per original length

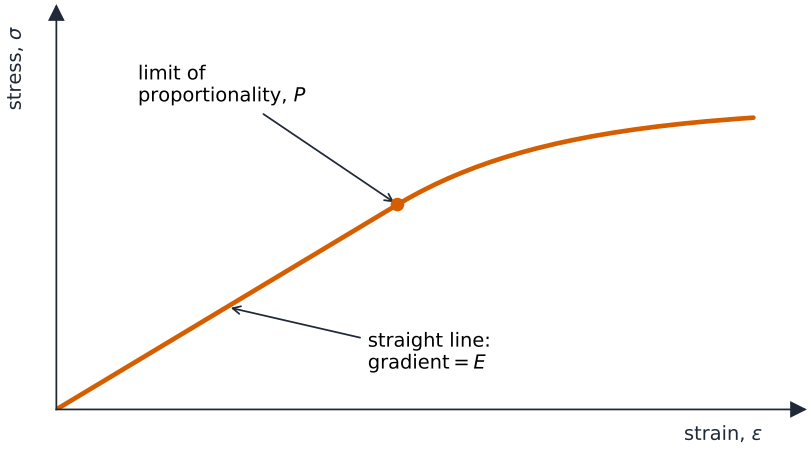
The **Young modulus** is the ratio of stress to strain in the Hooke's-law region:

$$E = \frac{\sigma}{\varepsilon} = \frac{F/A}{x/L_0} = \frac{FL_0}{Ax}.$$

Unit: Pa (about 10^{11} for metals; e.g. steel $\approx 2.0 \times 10^{11}$ Pa).

> **Worked example.** A steel wire of length 2.0 m and cross-sectional area 1.5×10^{-7} m^2 stretches by 1.0 mm under a load of 15 N. Find the Young modulus.

$$E = \frac{FL_0}{Ax} = \frac{15 \times 2.0}{(1.5 \times 10^{-7})(1.0 \times 10^{-3})} = 2.0 \times 10^{11} \text{ Pa}.$$



A stress–strain graph, straight up to the limit of proportionality P

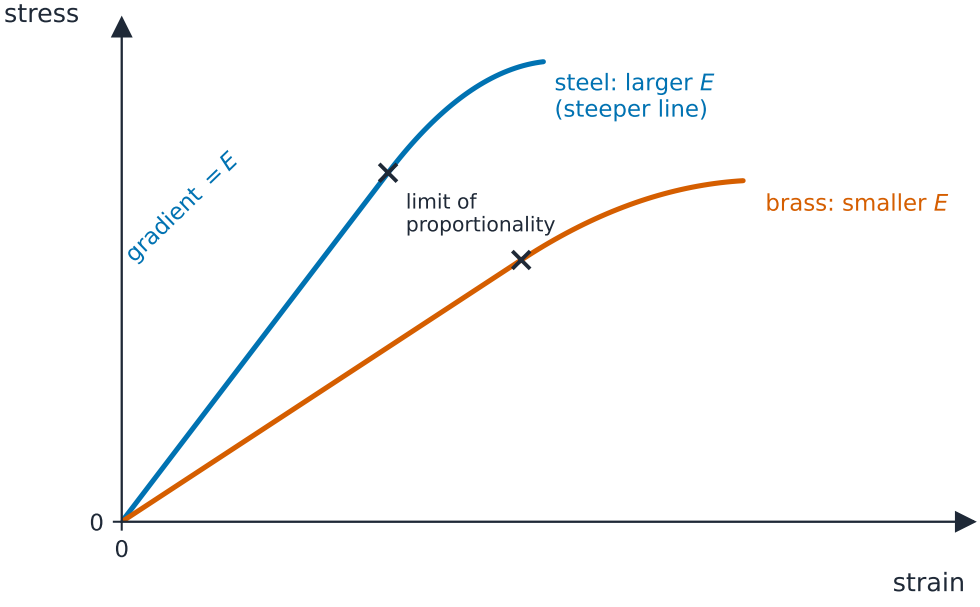
The Young modulus is a property of the **material** – it does not depend on the wire's shape or size. The spring constant k depends on both the material and the size: $k = EA/L_0$.

For the one-mark definition: **the Young modulus is the ratio of stress to strain** (for a material deformed within its limit of proportionality). Because it belongs to the material, a thicker wire of the same metal has the **same** Young modulus: it stretches less under the same load because its area is larger, not because the metal is stiffer. A "show that $k = EA/L_0$" question wants two lines: $k = F/x$ and $E = FL_0/(Ax)$, so $E = kL_0/A$. Two things follow: a wire's spring constant stays **constant** while Hooke's law holds, whatever the force, and a wire of the same metal with twice the diameter has four times the spring constant.

> **Worked example.** A copper wire of length 1.7 m and diameter 0.64 mm has Young modulus 1.2×10^{11} Pa. Find its spring constant.

$A = \pi(0.32 \times 10^{-3})^2 = 3.2 \times 10^{-7}$ m^2, so $k = EA/L_0 = 1.2 \times 10^{11} \times 3.2 \times 10^{-7}/1.7 = 2.3 \times 10^4$ N m^{-1}.

On a stress–strain graph the gradient of the straight part is E, so of two materials drawn on the same axes the **steeper** line is the stiffer material, and a wire with double the Young modulus is drawn as a line through the origin with twice the gradient. On a force–extension graph the gradient is EA/L_0 and the Young modulus is the gradient multiplied by L_0/A; only a stress–strain graph has a gradient equal to E itself.



Two metals on one stress–strain graph: the steeper line has the larger Young modulus

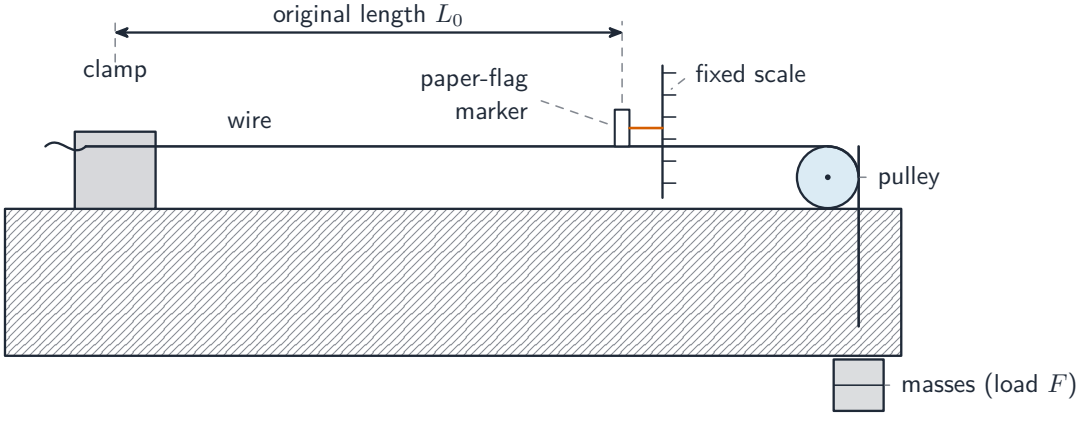
■ Experiment to find the Young modulus of a metal wire

A standard setup:

1. Clamp one end of a long, thin wire to a fixed support. Pass the wire over a **pulley** at the edge of the bench so it hangs straight down.
2. Measure the original length L_0 between the clamp and a marker near the pulley, using a metre rule.
3. Measure the **diameter** d of the wire at several places with a **micrometer** and take the average. Work out $A = \pi d^2/4$.
4. Hang weights one at a time. Record the load F and the extension x (how far the marker moves against a fixed scale).
5. Plot F against x. In the straight region the gradient is EA/L_0, so $E = \text{gradient} \times L_0/A$.

Why a **long, thin** wire? To make the extension big enough to measure well. Why repeat readings and measure d at several places? To reduce **random error** and check the wire is uniform.

A "describe an experiment" answer earns its marks for the **quantities measured and how** (L_0 with a metre rule; d with a micrometer at several points; the load from the masses or a newton meter; the extension from a marker read against a fixed scale, or a vernier scale), the **graph** (F against x, gradient EA/L_0, or stress against strain, gradient E) and the **precautions**: safety goggles in case the wire snaps, a small load first to straighten kinks, and readings taken on unloading as well as loading to check the wire stayed elastic.



Apparatus for measuring the Young modulus of a wire

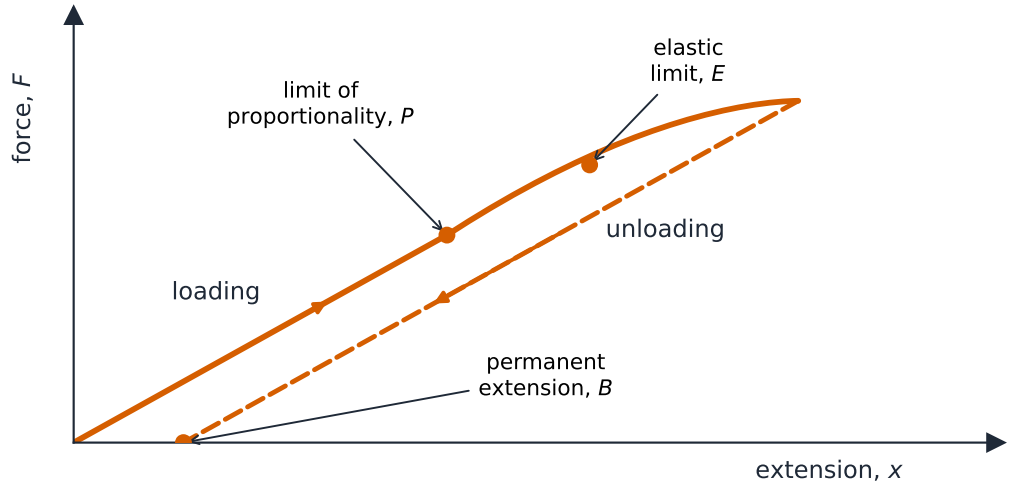
Elastic and plastic behaviour

As the load grows:

1. **Elastic and straight (Hooke obeyed)** – up to the limit of proportionality. Removing the load returns the object to its first length.
2. **Elastic but curved** – between the limit of proportionality and the **elastic limit** . The extension is no longer straight in the load, but on unloading the object still returns to its first length.
3. **Plastic** – past the elastic limit. On unloading, the object does **not** return to its first length; a permanent extension stays.

Hooke's law only holds in the straight, elastic region.

In words: **elastic deformation** means the object returns to its original length (or shape) when the load is removed; **plastic deformation** means it does not, and a permanent extension remains; the **elastic limit** is the maximum load (or extension) for which the deformation is still elastic; the **limit of proportionality** is the point beyond which the extension is no longer proportional to the load. On a graph the limit of proportionality is where the line stops being straight. The elastic limit lies a little beyond it and cannot be read from a loading line alone. So in a "which statement must be correct" item, the end of the straight part *is* the limit of proportionality, but whether a later point is the elastic limit or the breaking point cannot be told from the shape of the loading line.



Hooke up to limit of proportionality · plastic beyond yield

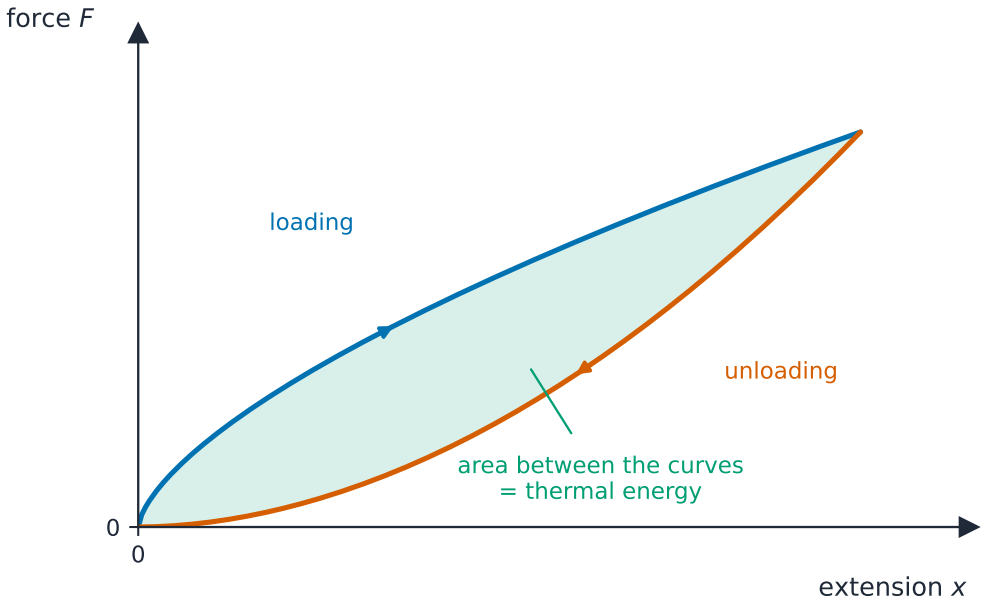
Force–extension past the elastic limit: P and E marked, with a permanent extension B left after unloading



A modern universal (tensile) testing machine stretches a sample and records the force and extension

On a force–extension graph for a material taken into the plastic region and then unloaded, the loading line and the unloading line are different. The unloading line is parallel to the first Hooke line but shifted to the right (the permanent extension left when the load reaches zero). The area between the loading and unloading lines is the energy turned into **thermal energy** in the material.

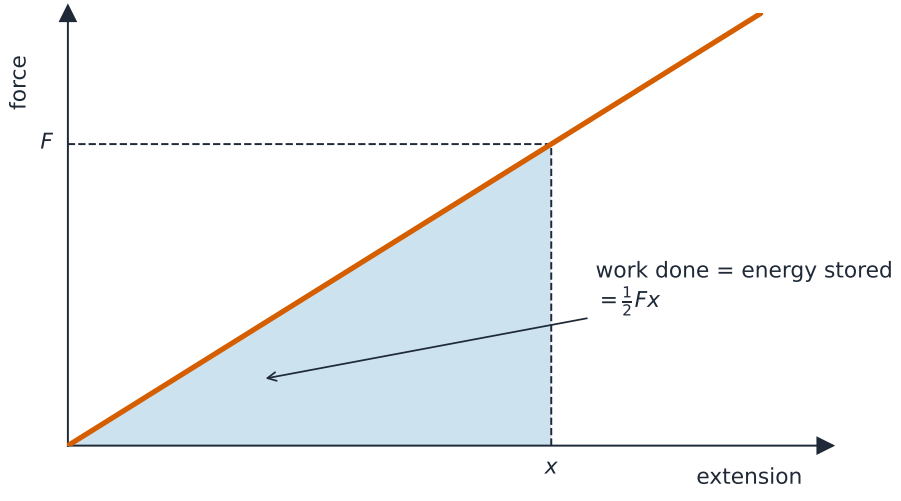
A **rubber band** also has different loading and unloading curves, but it returns to its original length: its deformation is elastic. The area between the two curves is again energy dissipated as thermal energy (**elastic hysteresis**), and the energy recovered on unloading is the area under the lower curve. The two-mark "explain why the work done in stretching the wire is not equal to the energy recovered when the force is removed" answer says that the wire was taken past its elastic limit, so part of the deformation is plastic and a permanent extension remains, and that part of the work done became thermal energy in the wire.



A rubber band returns to its original length, but the area between loading and unloading is energy lost as thermal energy

Energy stored in a stretched material

The **work done** in stretching a material from 0 to extension x, as the load grows from 0 to F, is the **area under the force–extension graph**.



$E = \frac{1}{2}Fx = \frac{1}{2}kx^2$ (area under F–x graph)

The work done stretching a material is the area under the force–extension graph

Why the area: work done is force multiplied by the distance moved, but here the

force grows as the material stretches, so the work is the sum of $F\,\Delta x$ over many small extensions, which is the area under the line. Within the limit of proportionality the force rises uniformly from 0 to F, so the average force is $\frac{1}{2}F$ and the work is $\frac{1}{2}Fx$. On a force–extension graph the gradient is the spring constant.

■ Hooke's-law material

When Hooke's law holds, the F against x graph is a straight line through the origin. The area under it from 0 to x is a triangle:

$$E_{\mathrm{P}} = \tfrac{1}{2}Fx = \tfrac{1}{2}kx^2.$$

This is the **elastic potential energy** stored in a spring or wire stretched within its limit of proportionality. An equal form:

$$E_{\mathrm{P}} = \frac{F^2}{2k}.$$

> **Worked example.** A spring of spring constant 50 N m^{-1} is stretched by 0.20 m, within its limit of proportionality. Find the elastic potential energy stored.

$$E_{\mathrm{P}} = \tfrac{1}{2}kx^2 = \tfrac{1}{2} \times 50 \times 0.20^2 = 1.0 \text{ J}.$$

> **Worked example.** A wire of spring constant 2.0×10^4 N m^{-1} is already extended by 2.0 mm. Find the work done to increase its extension to 3.0 mm.

The stored energy rises from $\frac{1}{2}kx_1^2$ to $\frac{1}{2}kx_2^2$:

$$\begin{aligned} W &= \tfrac{1}{2} \times 2.0 \times 10^4 \\ &\quad \times \left[(3.0 \times 10^{-3})^2 - (2.0 \times 10^{-3})^2\right] \\ &= 0.050 \text{ J} \end{aligned}$$

This is the trapezium under the line between the two extensions, not $\frac{1}{2}k(x_2 - x_1)^2$. Because $E_{\mathrm{P}} \propto x^2$ at fixed k, doubling the extension of a wire stores four times the energy (0.65 J becomes 2.6 J), and a stored energy gives the extension as $x = \sqrt{2E_{\mathrm{P}}/k}$: a spring of $k = 400$ N m^{-1} storing 0.32 J is compressed by $\sqrt{2 \times 0.32/400} = 0.040$ m. For two wires joined end to end the tension is the same in both, so the total energy stored is $\frac{1}{2}Fx_1 + \frac{1}{2}Fx_2$ with each wire's own extension.

■ Non-Hooke material

For a graph that is not a straight line (a stretched rubber band, or a spring past its limit of proportionality), find the area by counting grid squares or by using **trapezia**. The same idea holds: **the area under the force–extension graph is the work done on the material**. To estimate the work done up to the breaking point, count the squares under the whole curve (part squares as halves) and multiply by the

energy one square represents, the force step multiplied by the extension step; saying that the area was found by counting squares is the "explain your reasoning" mark.

■ Comparing stored energy

A common multiple-choice case: two materials are stretched by the same force, or by the same extension. Using $E_P = \frac{1}{2}Fx$:

- same F, smaller k (less stiff) $\rightarrow$ larger x $\rightarrow$ more energy stored.
- same x, larger k (stiffer) $\rightarrow$ larger F $\rightarrow$ more energy stored.

A sketch of E_P against extension or compression is a curve through the origin that gets steeper ($E_P \propto x^2$), not a straight line. Of two wires of the same length and area under the same load, the one with the smaller Young modulus extends more and so stores more energy.

When a stretched spring is released onto a mass, the elastic potential energy becomes **kinetic energy** (and gravitational potential energy if the mass rises). Set $\frac{1}{2}kx^2$ equal to $\frac{1}{2}mv^2$ (plus any mgh) to find the speed or height.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
Hooke's law	the extension is directly proportional to the applied force, provided the limit of proportionality is not exceeded
spring constant	the force per unit extension
limit of proportionality	the point beyond which the extension is no longer proportional to the applied force
elastic limit	the maximum force (or extension) for which the material returns to its original length when the force is removed
elastic deformation	the material returns to its original length (shape) when the force is removed
plastic deformation	the material does not return to its original length when the force is removed; a permanent extension remains
stress	the force per unit cross-sectional area
strain	the extension per unit original length
Young modulus	the ratio of stress to strain (within the limit of proportionality)

Term	Definition
elastic potential energy	the energy stored in an object because it has been stretched or compressed

Exam tips

- **Hooke's law ($F = kx$)** holds only up to the limit of proportionality.
- **Stress** $= F/A$**, strain** $= x/L$**, Young modulus = stress/strain** (gradient of the straight part of the stress-strain graph) – watch the units (Pa).
- Energy stored = **area under the force-extension graph** $= \frac{1}{2}Fx$ in the elastic region.
- Distinguish **elastic** (returns to shape) from **plastic** (permanent) deformation.

Common mistakes

- Using the diameter as the radius in $A = \pi r^2$, or leaving an area in mm^2. Halve the diameter first; $1\ \text{mm}^2 = 10^{-6}\ \text{m}^2$.
- Reading the work done off a length–force graph as the whole area down to the axis. Only the triangle above L_0 is work.
- Treating stored energy as proportional to extension. It goes as x^2: double the extension, four times the energy; and the work done between two extensions is the difference of two $\frac{1}{2}kx^2$ values.
- Stating Hooke's law without its condition. "Provided the limit of proportionality is not exceeded" is part of the law.
- Swapping the limit of proportionality (the end of the straight line) and the elastic limit (the end of elastic behaviour, a little beyond it).
- Saying a thicker wire has a larger Young modulus. The modulus is the material's; the thicker wire has a larger spring constant.

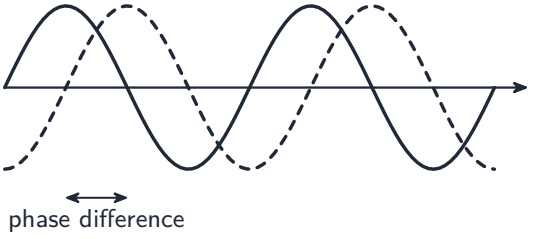
Waves

A-Level Physics

Progressive waves



Ripples spreading on water are progressive waves that carry energy outward.



Two waves of the same frequency, shifted by a phase difference

A **wave** carries **energy** from one place to another without moving matter overall. The particles of the **medium** **oscillate** about fixed rest positions; only the disturbance (and its energy) **propagates** . Examples: a **transverse wave** on a rope, a **longitudinal wave** on a slinky spring, ripples on water, and sound in air. A wave that travels and carries energy is a **progressive wave** .

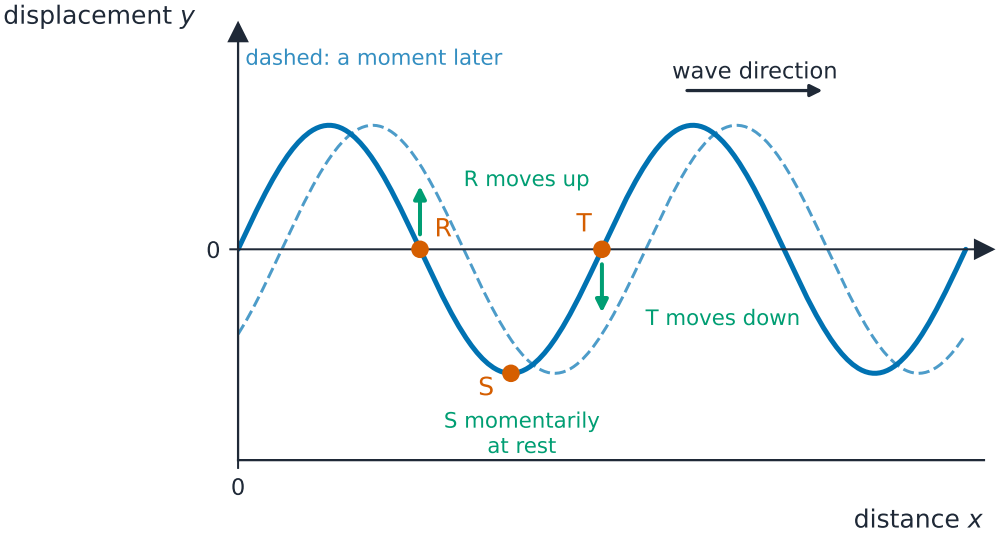
Wave motion is a series of oscillations of the particles of a medium, each passing the disturbance on to the next: shake one end of a rope, push one end of a spring, or touch the surface of the water in a **ripple tank** , and the oscillation travels away from where it started. For the one-mark definition, a progressive wave transfers energy from one place to another without transferring matter; the particles only oscillate about their rest positions.

Key terms

- **displacement** y – how far a particle has moved from its rest position at a moment. A **vector** .
- **amplitude** A – the largest displacement from the rest position.
- **wavelength** λ – the shortest distance along the wave between two points that move **in phase** (for example, two next-door **crests**).
- **period** T – the time for one full oscillation of a particle.
- **frequency** f – the number of full oscillations per second; $f = 1/T$. Unit: **hertz** , Hz.
- **speed** v – how fast a crest travels along the medium.
- **phase difference** – the fraction of a cycle by which one oscillation leads or lags another. Given in **radians** (a full cycle is 2π) or degrees (a full cycle is 360°).

Two points one wavelength apart are in phase (phase difference 0 or 2π). Two points half a wavelength apart are exactly out of phase (phase difference π).

To find the phase difference between two points from a displacement–distance graph, divide their separation by the wavelength and multiply by 360° (or 2π): two points 0.50 m apart on a wave of wavelength 2.0 m ($v = 600$ m s^{-1}, $f = 300$ Hz) differ in phase by $0.25 \times 360° = 90°$. Only the fraction of a cycle matters, so 450° is the same as 90°. From two displacement–time graphs, read the time shift between the peaks as a fraction of the period. Which way is a point moving at the instant a graph shows? The whole profile moves along, so each point takes up the displacement that its neighbour on the side the wave comes from has now; a point at a crest or a trough is momentarily at rest.



Which way each point moves: towards the displacement its neighbour on the incoming side has now; a crest or trough is momentarily at rest

Worked example. Points R and T on a string are 0.62 cm apart and a quarter of a cycle out of phase. The wave speed is 0.27 m s^{-1}. Find the frequency.

A quarter of a cycle is a quarter of a wavelength, so $\lambda = 4 \times 0.62 = 2.5$ cm and $f = v/\lambda = 0.27/0.025 = 11$ Hz.

The wave equation

In one period T, the wave moves forward by one wavelength λ. So speed = distance/time = $\lambda/T = \lambda f$:

$$v = f\lambda.$$

This comes straight from the definitions of speed, frequency and wavelength, and works for every progressive wave. For the two-mark derivation write both steps: in one period T the wave travels one wavelength λ, so $v = \lambda/T$; and $f = 1/T$, so $v = f\lambda$. Because v is fixed by the medium, a higher frequency means a shorter wavelength. A period question is the same equation the other way round: light of wavelength 460 nm has $f = c/\lambda = 6.5 \times 10^{14}$ Hz and $T = 1/f = 1.5 \times 10^{-15}$ s.

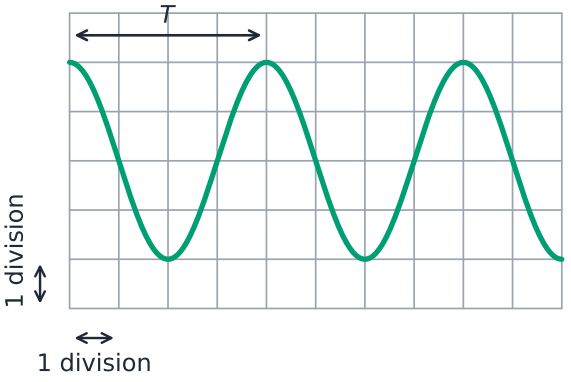
Reading a CRO trace

A **cathode-ray oscilloscope** (CRO) draws a voltage signal – for sound, the output of a microphone – against time. Two controls matter:

- **time-base** (seconds per division across): turns horizontal distance on the screen into time. Read the period T as the distance between two next-door peaks, then $f = 1/T$.
- **y-gain** (volts per division up): turns vertical distance into voltage. The amplitude in volts is the peak height from the centre line.

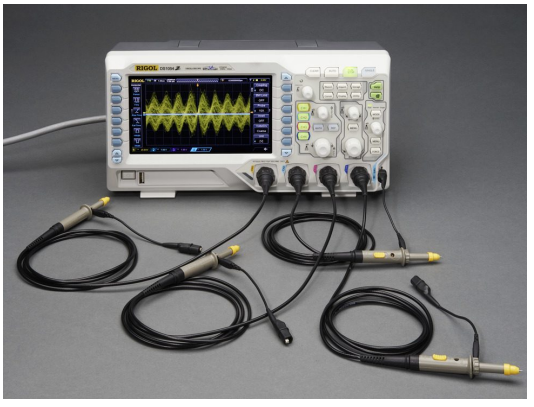
If the time-base is 5 ms/div and one full cycle takes 4 divisions, then $T = 4 \times 5$ ms = 20 ms and $f = 50$ Hz.

The time-base is the time represented by one division (or one centimetre) across the screen, in s div^{-1}. To set it for a given signal, work backwards: a 2000 Hz sound has $T = 0.50$ ms, so for one cycle to span 2.5 cm the time-base must be $0.50/2.5 = 0.20$ ms cm^{-1}. For the amplitude, multiply the peak height in divisions by the y-gain: 2.0 divisions at 3.5 mV cm^{-1} is 7.0 mV. When the intensity of the sound is reduced to a quarter at the same frequency, the trace keeps its period and its peaks halve in height, because $I \propto A^2$.



period $= n \times$ timebase $\cdot$ peak voltage $= n \times$ Y-gain

Reading the period T from a CRO trace using the grid and time-base



A real oscilloscope: the grid lets you read off the period and the amplitude

Intensity of a wave

A wave carries energy. The **intensity** at a point is the **power** passing through unit area at right angles to the direction of travel:

$$I = \frac{P}{A}.$$

Unit: W m^{-2}.

Intensity is proportional to the square of the amplitude:

$$I \propto A^2.$$

For a **point source** sending out energy equally in all directions, the **wavefronts** are spheres; the surface area at distance r is $4\pi r^2$, so

$$I = \frac{P}{4\pi r^2}, \qquad I \propto \frac{1}{r^2}.$$

Doubling the distance cuts the intensity to a quarter, which means the amplitude is halved (since $I \propto A^2$).

> **Worked example.** A lamp emits 60 W of light equally in all directions. Find the intensity of the light 2.0 m away.

$$I = \frac{P}{4\pi r^2} = \frac{60}{4\pi(2.0)^2} \approx 1.2 \text{ W m}^{-2}.$$

> **Worked example.** Light of power 750 W falls at right angles on a square solar panel of side 1.2 m. Find the intensity.

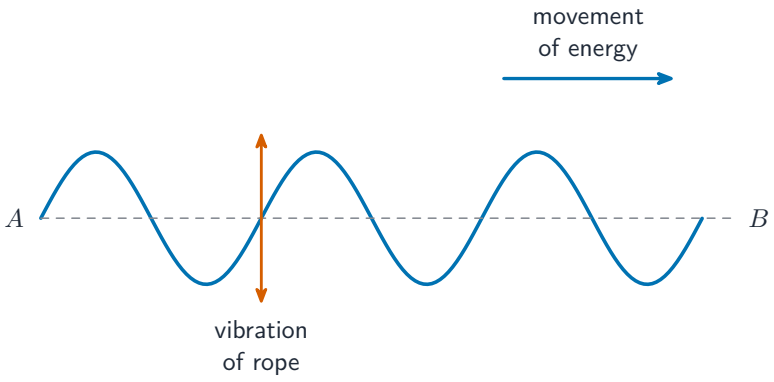
$I = P/A = 750/1.2^2 = 520$ W m^{-2}. A smaller panel in the same light receives the same intensity but less power, in proportion to its area. The other way round, a magnifying glass of radius r collects a power $I \times \pi r^2$ and concentrates it on a small spot.

Two rules the multiple-choice questions turn on. Because $I \propto A^2$, the ratio of two intensities is the **square** of the ratio of the amplitudes: waves of amplitude 3.0 cm and 2.0 cm have intensities in the ratio 2.25, and when an amplitude falls to a half the intensity falls to a quarter. For a point source, $I \propto 1/r^2$ means $A \propto 1/r$, so a sketch of A/A_0 against d/x_0 is a curve falling as $1/d$, not a straight line. The one-line statement the scheme wants: intensity is proportional to the amplitude squared.

Transverse and longitudinal waves

■ Transverse waves

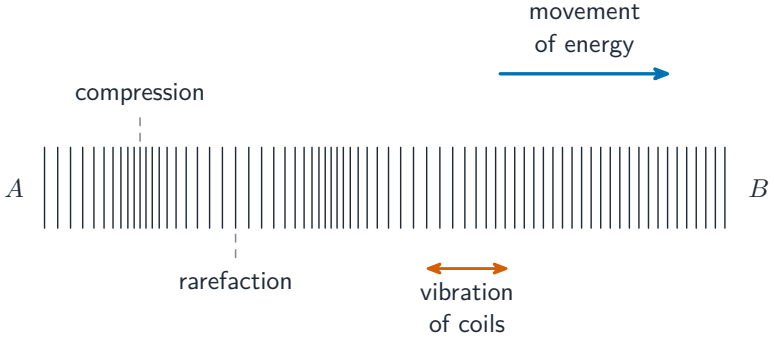
The particles oscillate **perpendicular** to the direction the energy travels. A wave on a rope, all electromagnetic waves, and S-waves in the Earth are transverse.



Transverse wave on a rope

Longitudinal waves

The particles oscillate **parallel** to the direction the energy travels. Sound in any medium, P-waves in the Earth, and the squashes on a slinky are longitudinal. The wave is made of **compressions** (higher pressure, particles close together) and **rarefactions** (lower pressure, particles spread out).



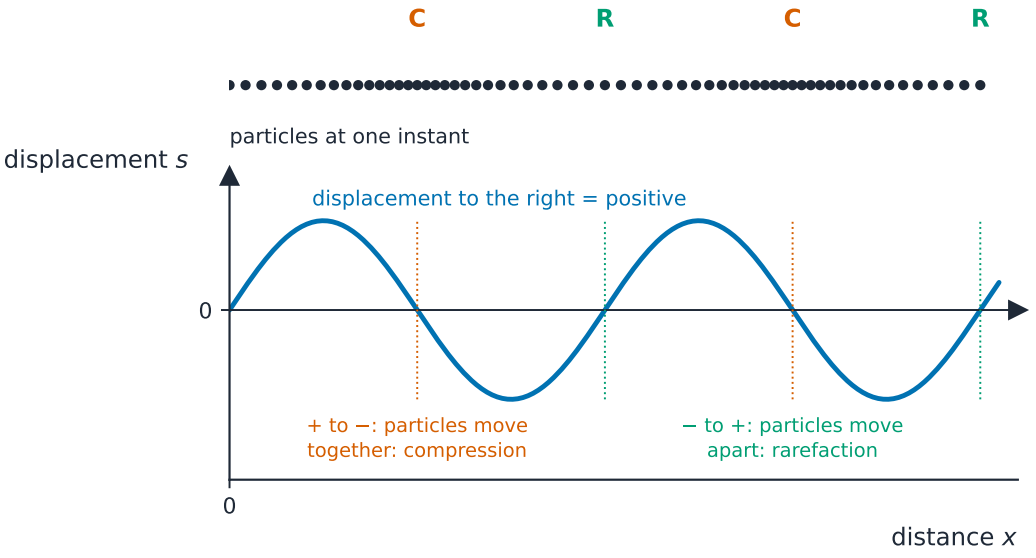
Longitudinal wave on a slinky spring

Graphs of waves

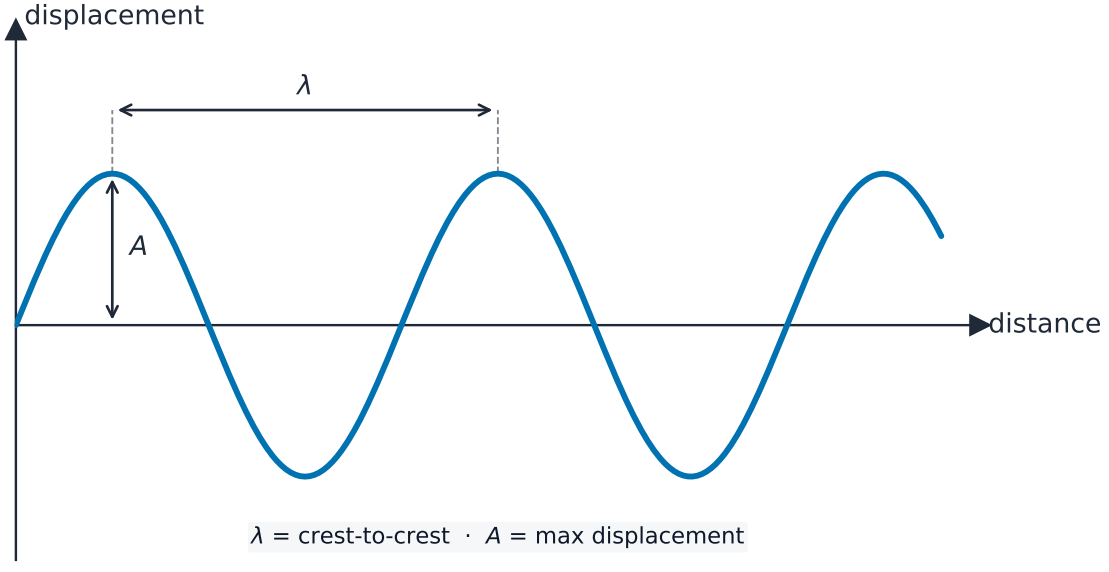
A graph of **particle displacement against position** at one moment looks like a **sine curve** for both kinds of wave. The difference: for a transverse wave the displacement axis is the real sideways displacement; for a longitudinal wave it is the small back-and-forth displacement along the direction of travel (positive one way, negative the other).

For a longitudinal wave take displacement to the **right** as positive. Where the graph crosses zero going from positive to negative, the particles on either side have moved **towards** that point, so it is a compression; where it crosses from negative to positive they have moved apart, a rarefaction. Two neighbouring compressions are one wave-

length apart, and a compression and the next rarefaction are half a wavelength apart. The direction of motion of a particle follows the same rule as for a transverse wave: it moves towards the displacement of its neighbour on the side the wave comes from.



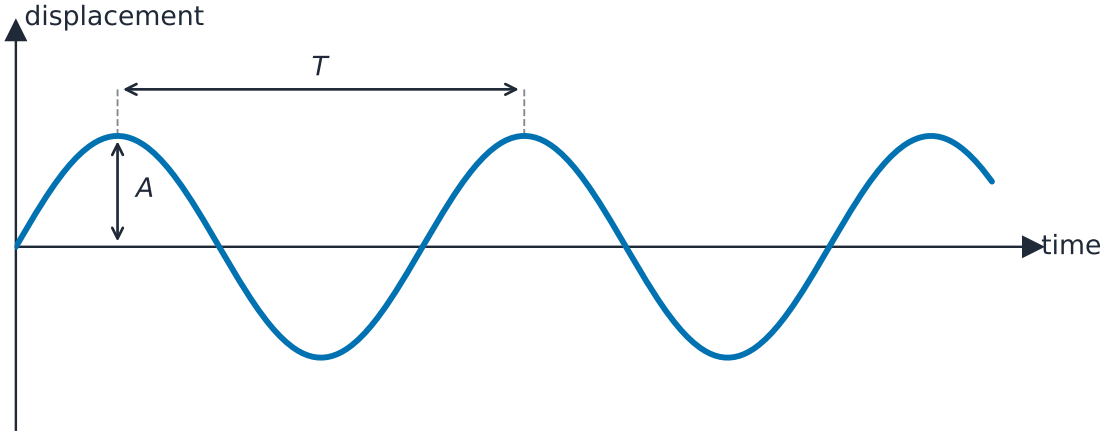
The same longitudinal wave as particles and as a graph: a compression where the displacement changes from positive to negative, a rarefaction where it changes from negative to positive



A displacement–distance graph shows the wave's amplitude and wavelength

A graph of **particle displacement against time** at one point in space is also a sine

curve for both kinds. Read the period T from this graph.



amplitude A = peak displacement · period T = time for one full cycle

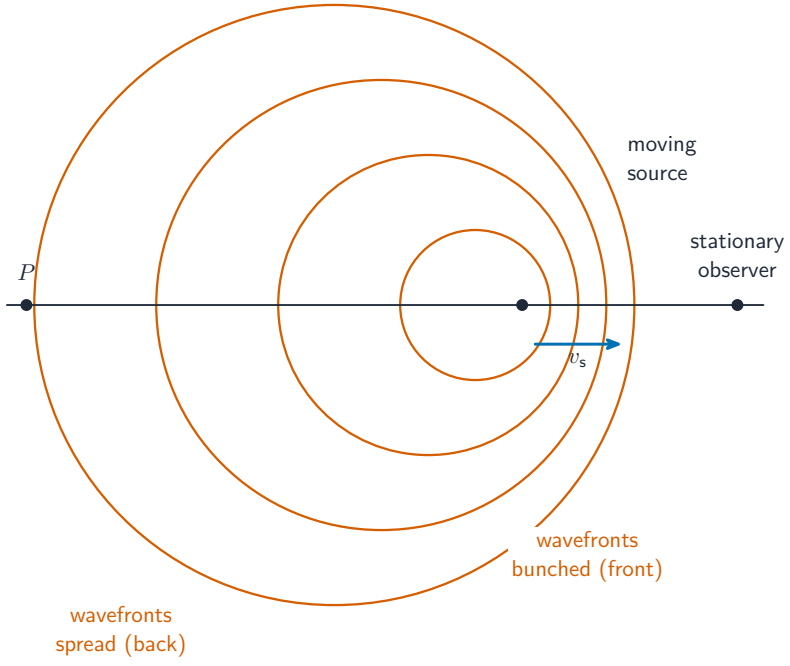
A displacement–time graph shows the wave's amplitude and period

The two-mark comparison, with reference to the direction of energy transfer: in a transverse wave the oscillations are **perpendicular** to the direction of energy transfer; in a longitudinal wave they are **parallel** to it. Both kinds transfer energy without transferring matter, both can be reflected, refracted and diffracted, and both can form stationary waves (topic 8). Only transverse waves can be polarised, and only electromagnetic waves, which are all transverse, can travel through a vacuum; sound needs a medium. Reading the two graphs together: the displacement–distance graph gives λ, the displacement–time graph gives T, and $v = \lambda/T$.

Doppler effect (moving source, stationary observer)

When the **source** of a sound moves relative to a **stationary observer**, the heard frequency is different from the source frequency. This is the **Doppler effect**.

- source moving towards the observer: the wavefronts in front are squashed, so the wavelength is shorter and the heard frequency is **higher**.
- source moving away from the observer: the wavefronts behind are spread out, so the wavelength is longer and the heard frequency is **lower**.



A moving source squashes the wavefronts ahead of it, raising the observed frequency

The formula (source moving at speed v_s along the line to the observer; wave speed v, source frequency f_s, heard frequency f_o):

$$f_o = \frac{v \cdot f_s}{v \pm v_s}.$$

Choose the sign to match the physics:

- minus sign on the bottom when the source moves **towards** the observer ($f_o > f_s$),
- plus sign when the source moves **away** ($f_o < f_s$).

You only need the case of a stationary observer.

> **Worked example.** A car horn at $f_s = 800$ Hz moves at 30 m s^{-1} towards a still listener. Speed of sound $v = 340$ m s^{-1}:
>
> $$f_o = \frac{340 \times 800}{340 - 30} = \frac{272\,000}{310} \approx 877 \text{ Hz}.$$

Moving away at the same speed, the listener hears $340 \times 800/(340 + 30) = 735$ Hz: the frequency is constant while the car approaches, drops as it passes, and is constant again, lower, as it recedes. A sketch of observed frequency against time for a source passing at constant speed is two level lines joined by a fall. Working backwards, a source of 1200 Hz heard as 960 Hz is moving away ($f_o < f_s$): $960 = 1200 \times 340/(340 + v_s)$ gives $v_s = 85$ m s^{-1}. A buzzer swung in a horizontal circle at 25 m s^{-1} while emitting 846 Hz is heard between $846 \times 330/355 = 786$ Hz (moving directly away)

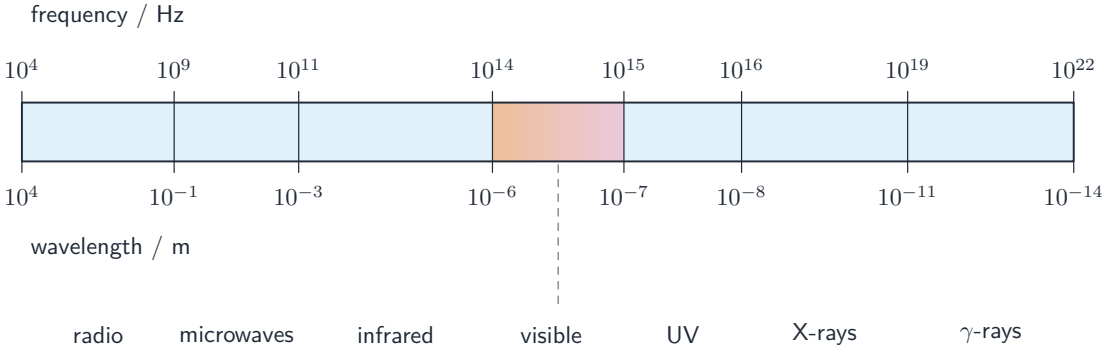
and $846 \times 330/305 = 915$ Hz (moving directly towards), once each per revolution. If the period of a CRO trace of the sound rises continuously, the observed frequency is falling: the source is moving away with increasing speed. The Doppler effect happens for all waves, light included (topic 25); only the sound formula is examined here.

Electromagnetic spectrum

All **electromagnetic waves** (EM waves) are **transverse** and travel in a **vacuum** at the same speed:

$$c = 3.00 \times 10^8 \text{ m s}^{-1}.$$

The **electromagnetic spectrum** includes radio waves, **microwaves**, **infrared**, visible light, **ultraviolet**, **X-rays** X and γ**-rays** γ.



The electromagnetic spectrum

Approximate wavelength ranges in free space (learn the orders of magnitude):

- radio waves: $> 10^{-1}$ m (up to many km).
- microwaves: 10^{-3} m to 10^{-1} m.
- infrared: $\sim 7 \times 10^{-7}$ m to 10^{-3} m.
- visible light: 400 nm (violet) to 700 nm (red), i.e. 4×10^{-7} m to 7×10^{-7} m.
- ultraviolet: $\sim 10^{-8}$ m to 4×10^{-7} m.
- X-rays: $\sim 10^{-11}$ m to 10^{-8} m.
- γ-rays: $< 10^{-11}$ m.

The boundaries between regions are not sharp. Use $c = f\lambda$ to change between wavelength and frequency. Only light with wavelengths 400–700 nm can be seen.

To identify a region, convert to a wavelength with $\lambda = c/f$ and compare with the ranges: 2.1 cm is a microwave; 138 pm $= 1.4 \times 10^{-10}$ m is an X-ray; 30 THz gives $\lambda = 1.0 \times 10^{-5}$ m, infrared; 3.0×10^{16} Hz gives 1.0×10^{-8} m, ultraviolet. Visible light spans frequencies of about 4.3×10^{14} to 7.5×10^{14} Hz, so a wave of 5.0×10^{14} Hz can

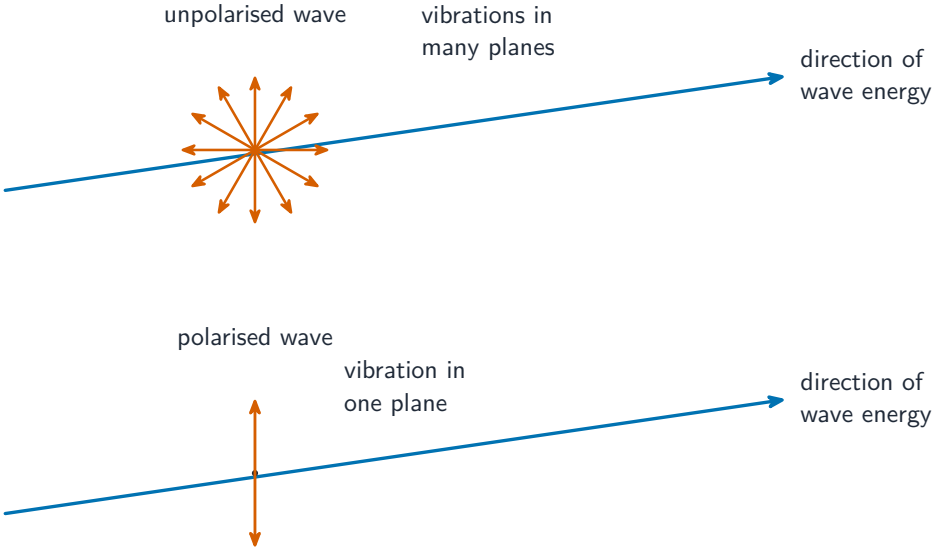
be seen and one of wavelength 5.0×10^{-6} m (infrared) cannot. Red light has a longer wavelength and a lower frequency than green. A list "in order of increasing wavelength" runs γ-rays, X-rays, ultraviolet, visible, infrared, microwaves, radio waves. A pulse of light reflected from a wall 150 m away returns after $2 \times 150/(3.00 \times 10^8) = 1.0$ μs.

Polarisation

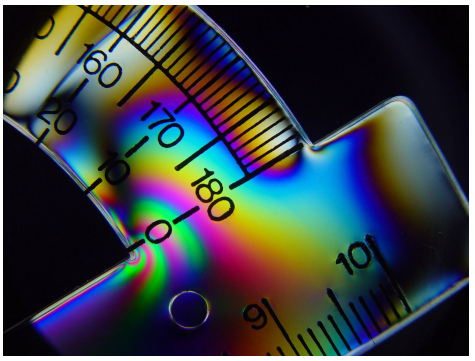
Polarisation means making a transverse wave oscillate in one plane only.

- Only **transverse** waves can be polarised – the oscillation is perpendicular to the direction of travel, so different perpendicular planes are real choices.
- **Longitudinal** waves (sound) cannot be polarised – the oscillation is along the direction of travel, so there is no other plane.

So polarisation is a test: if a wave can be polarised, it must be transverse.



Unpolarised waves vibrate in many planes; a polarised wave vibrates in one plane



A see-through plastic protractor placed between two crossed polarising filters. Light only reaches your eye because the stressed plastic rotates its plane of polarisation – the colours map where the plastic is squeezed most. With ordinary light it would just look clear

Malus's law

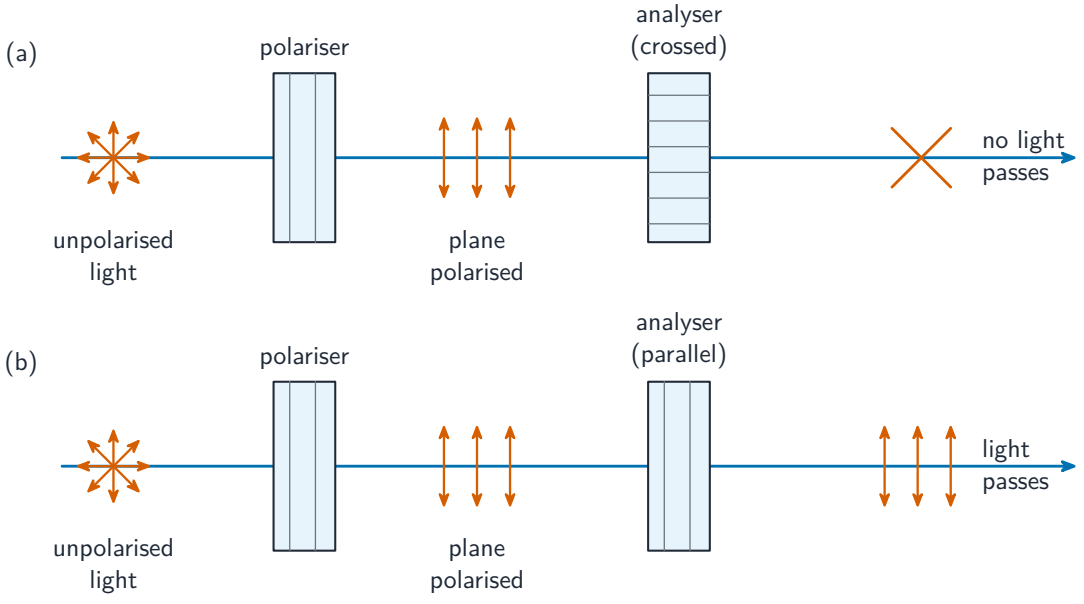
Plane-polarised light of intensity I_0 passes through a **polarising filter** whose **transmission axis** is at angle θ to the plane of polarisation. The transmitted intensity is given by **Malus's law**:

$$I = I_0 \cos^2 \theta.$$

- $\theta = 0°$: filter lined up with the polarisation, $I = I_0$, all passes through.
- $\theta = 90°$: filter at right angles, $I = 0$, all blocked.
- $\theta = 60°$: $I = I_0 \cos^2 60° = I_0 \cdot 0.25 = I_0/4$.

Worked example. Plane-polarised light of intensity 12 W m^{-2} meets a polarising filter whose axis is at 30° to the plane of polarisation. Find the transmitted intensity.

$$I = I_0 \cos^2 \theta = 12 \times \cos^2 30° = 12 \times 0.75 = 9.0 \text{ W m}^{-2}.$$



Crossed filters (a) block the light; parallel filters (b) let it pass

For two filters in a row, use Malus's law twice with the angle between each pair. Be careful with the angle each time – after the first filter the polarisation is along that filter's axis, and the second filter's angle is measured from there.

(You do not need to work out the effect of a polarising filter on an unpolarised wave.)

When a filter is rotated through 360° in front of plane-polarised light, the transmitted intensity varies as $\cos^2\theta$: it is a maximum when the transmission axis is parallel to the plane of polarisation (0° and 180°), zero when perpendicular (90° and 270°), and never negative. Because $I \propto A^2$, the transmitted **amplitude** is $A_0\cos\theta$: at 45° the intensity halves and the amplitude falls to 0.71 of its value. For two filters, the first sets the plane of polarisation along its own axis, so the second filter's angle is measured from that: vertically polarised light through filters at 50° and then 80° to the vertical keeps $\cos^2 50° \times \cos^2 30° = 0.31$ of its intensity. The one-mark definitions: polarisation is the oscillation of a transverse wave in a single plane, which contains the direction of travel; sound cannot be polarised because its oscillations are along the direction of travel, so there is no plane to select.

■ Reading wave motion over time

- A particle oscillating between $-A$ and $+A$ travels a total distance $4A$ **in one complete cycle**. Over n complete cycles, $D = 4nA$; this is particle travel, not wave travel.
- In n periods, the wave travels $s = n\lambda$ and the time is $t = nT = n/f$.
- After $T/4$, a progressive-wave profile has moved $\lambda/4$ in its propagation direction. A stationary-wave profile changes height without moving its nodes: do not shift it sideways.

- If a polariser starts perpendicular to the incident polarisation and rotates by ϕ, its angle to the polarisation is $90° - \phi$, so $I = I_0 \sin^2 \phi$.

> **Worked example.** A particle has amplitude 3.0 mm and completes four cycles. $D = 4nA = 4 \times 4 \times 3.0 = 48$ mm. A wave of frequency 25 Hz takes $t = n/f = 4/25 = 0.16$ s for those cycles.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
progressive wave	a wave that transfers energy from one place to another without transferring matter
transverse wave	a wave whose oscillations are perpendicular to the direction of energy transfer
longitudinal wave	a wave whose oscillations are parallel to the direction of energy transfer
displacement	the distance of a particle from its equilibrium (rest) position, in a stated direction
amplitude	the maximum displacement of a particle from its equilibrium position
wavelength	the minimum distance between two points on the wave that are in phase (for example, adjacent crests)
period	the time for one complete oscillation of a particle (or for the wave to travel one wavelength)
frequency	the number of oscillations per unit time (or the number of wavefronts passing a point per unit time)
phase difference	the fraction of a cycle, as an angle, by which one oscillation leads or lags another
intensity	the power per unit area, at right angles to the direction of travel
Doppler effect	the change in the observed frequency of a wave when the source moves relative to the observer
polarisation	the oscillations of a transverse wave are in one plane only, containing the direction of travel

Term	Definition
time-base	the time represented by one division across the screen of a CRO

Exam tips

- Define terms precisely (displacement, amplitude, wavelength, period, frequency) and use $v = f\lambda$.
- Distinguish **transverse** (vibration perpendicular to travel) from **longitudinal** (parallel); only transverse waves can be **polarised**.
- For the **Doppler effect**, the observed frequency **rises** as the source approaches and falls as it recedes.
- Learn the **electromagnetic spectrum** order; all its waves travel at c in a vacuum.

Common mistakes

- A phase difference bigger than one cycle, or degrees where radians were asked. Only the fraction of a cycle matters: 450° is 90°, and 90° is $\pi/2$.
- Reading the wavelength from a displacement–time graph. That graph gives the period; the wavelength comes from the displacement–distance graph.
- Doubling the amplitude and "doubling" the intensity. Intensity goes as the amplitude squared: four times.
- Choosing the Doppler sign by feel. Towards means minus on the bottom (frequency up); away means plus (frequency down). Check that the answer lies on the right side of f_s.
- Using the first filter's angle for the second. After a filter the light is polarised along that filter's axis, and the next angle is measured from there.
- Saying that sound can be polarised. Only transverse waves can.

Superposition

A-Level Physics

Principle of superposition

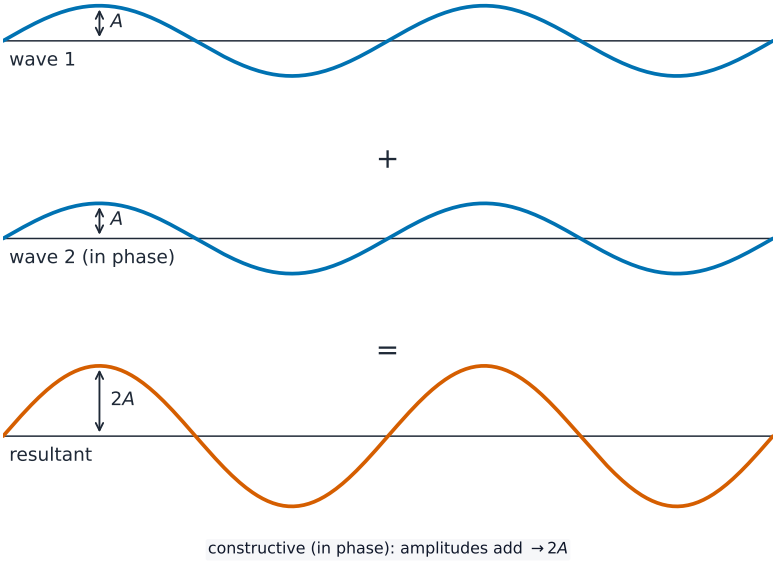
When two or more **waves** overlap at a point, the **displacement** there is the **vector sum** of the displacements each wave would make on its own. This is the **principle of superposition** .

The waves pass through each other and come out unchanged. Superposition is the base of everything in this topic.

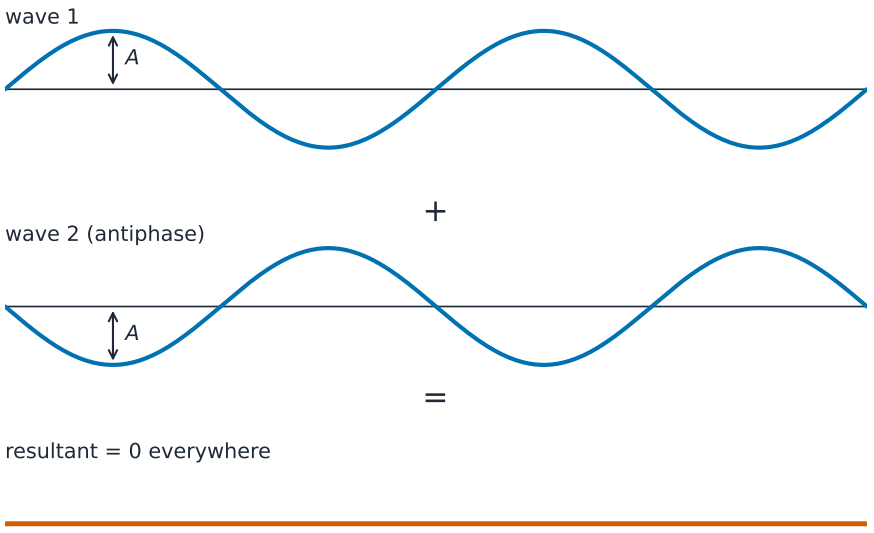
For the two-mark statement: **when two or more waves meet at a point, the resultant displacement is the sum of the displacements of the individual waves.** It is the displacements that add, not the amplitudes or the intensities, and the principle applies whenever waves of the same type overlap; the waves do not have to be coherent or of equal amplitude.

If two waves of **amplitude** A_1 and A_2 meet:

- **in phase** (**crest** meets crest): the amplitude is $A_1 + A_2$ (**constructive interference**).
- exactly out of phase (crest meets **trough** , **phase difference** π): the amplitude is $|A_1 - A_2|$ (**destructive interference**).
- any other phase difference ϕ: the amplitude is somewhere between these two.



Two waves arriving in phase add to give double the amplitude (constructive)



destructive (out of phase by λ/2): cancel if equal *A*

Two waves arriving exactly out of phase cancel to zero (destructive)

For **intensity** , $I \propto A^2$. Two equal waves meeting in phase give intensity $(2A)^2 = 4A^2$ – **four times** the intensity of one wave alone.

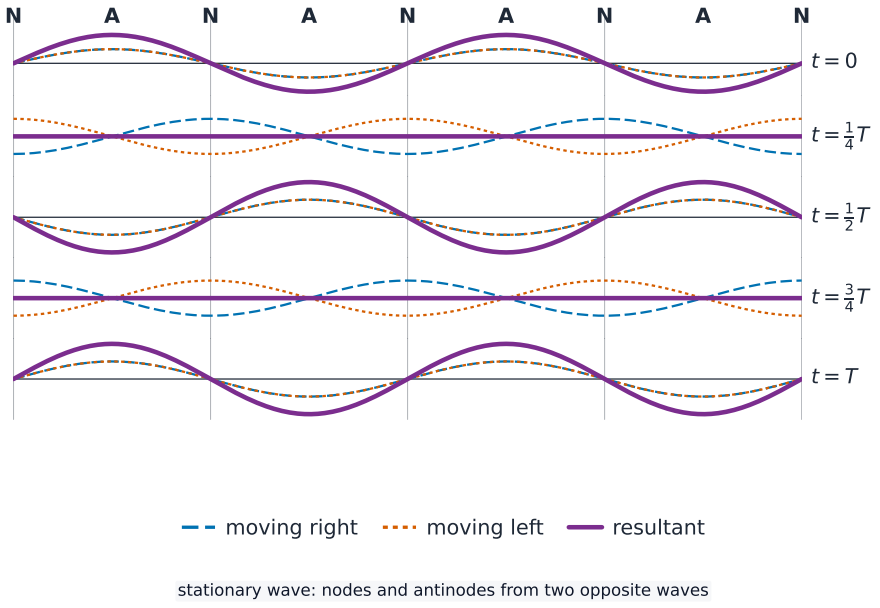
Because $I \propto A^2$, work in amplitudes first. Two waves of intensities I and $4I$ have amplitudes A and $2A$: superposing in phase gives amplitude $3A$ and intensity $9I$, in antiphase amplitude A and intensity I. So two coherent waves of different intensities never cancel completely: the minima have intensity $(A_2 - A_1)^2$, not zero. A wave of amplitude $2A$ meeting one of amplitude $A/2$ travelling the opposite way gives a resultant that varies between $2.5A$ and $1.5A$, a stationary pattern with no true nodes. Doubling the amplitude of one of two equal waves that meet in phase takes the resultant from $2A$ to $3A$, so the intensity rises from $4A^2$ to $9A^2$: 2.25 times.

Stationary (standing) waves

When two identical **progressive waves** travel in **opposite directions** and overlap, they make a **stationary wave** . Examples: a wave on a string **reflected** from a fixed end overlapping the incoming wave; sound in an air column reflected from a closed end; microwaves between an emitter and a metal sheet.

Explain how the stationary wave is formed (three or four marks, asked for a string, an air column and microwaves alike): the wave from the source travels to the far end (the wall, the closed end, the metal plate) and is **reflected**; the incident and reflected waves have the same frequency, wavelength and speed and travel in **opposite directions**; they **superpose** where they overlap; at the points where they always meet in phase the displacements add to give the maximum amplitude, an antinode, and at the points where they always meet in antiphase they cancel, a node. For "state the conditions": two waves of the same type, with the same frequency (and wavelength)

and speed, travelling in opposite directions along the same line; equal amplitudes are needed only for the nodes to have zero amplitude. The metal plate in the microwave experiment is there to reflect the waves back along their own path.



A stationary wave forms where two opposite waves overlap (N marks a node, A an antinode)

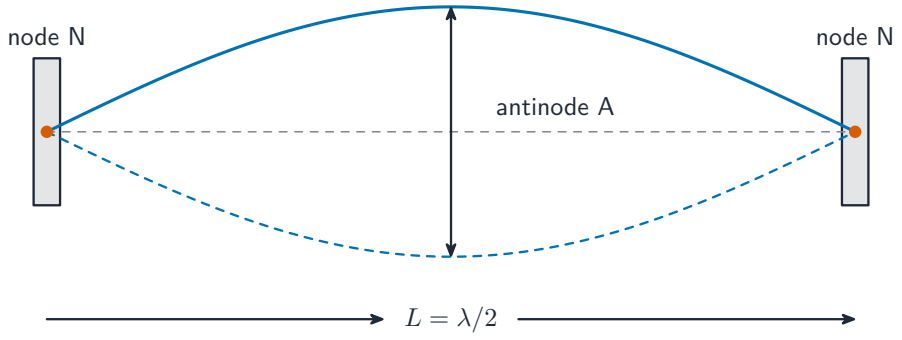
Nodes and antinodes

In a stationary wave:

- **node** – a point that is always at zero displacement (the two waves always cancel). The distance between next-door nodes is $\lambda/2$.
- **antinode** – a point of largest amplitude (the two waves always add). The distance between next-door antinodes is $\lambda/2$.
- a node and the next antinode are $\lambda/4$ apart.

Particles between two nodes oscillate **in phase** with each other, but with different amplitudes (largest at the antinode, zero at the nodes). Particles on opposite sides of a node oscillate in **antiphase** (phase difference π).

So the phase difference between two points on the same loop (between adjacent nodes) is 0°; between points on neighbouring loops it is 180°; and between points on loops separated by one whole loop it is 0° again. Every point reaches its maximum displacement at the same instant and passes through zero at the same instant. A sketch of the string a quarter of a cycle after the instant of maximum displacement is a straight line along the rest position; half a cycle later it is the mirror image of the first sketch. In one period a particle at an antinode travels four amplitudes, so a particle that moves 72 mm in one and a half periods has an amplitude of 12 mm.



Fundamental mode on a stretched string – one loop, with L equal to half a wavelength

> **Worked example.** A string of length 0.80 m is fixed at both ends and vibrates in its fundamental mode, where the wave speed is 240 m s^{-1}. Find the fundamental frequency.

One loop fits the string, so $\lambda = 2L = 1.60$ m. Then

$$f = \frac{v}{\lambda} = \frac{240}{1.60} = 150 \text{ Hz}.$$

How a stationary wave differs from a progressive wave: it does not carry **energy** along its length, the pattern does not move along, and the nodes stay fixed; a progressive wave has the same amplitude everywhere and carries energy. The three differences the scheme lists: a progressive wave transfers energy and a stationary wave does not; in a progressive wave every point has the same amplitude, in a stationary wave the amplitude varies from zero at a node to a maximum at an antinode; in a progressive wave neighbouring points differ in phase, in a stationary wave all the points between two nodes are in phase.

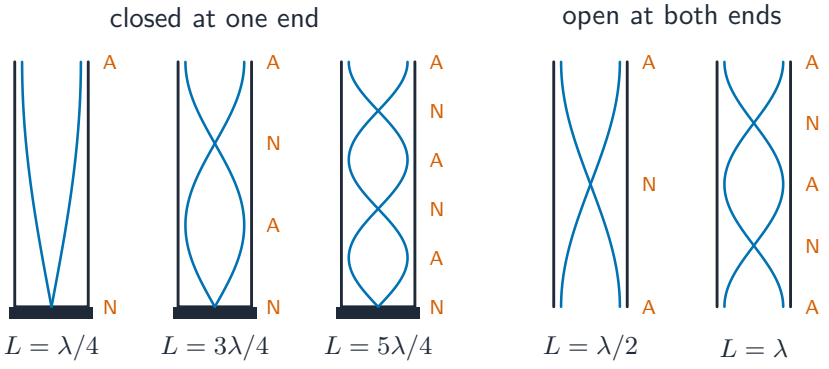
■ Measuring wavelength from node spacing

Drive a string with a vibrator at frequency f until a stationary pattern appears. Measure the distance between two well-separated nodes and divide by the number of half-**wavelengths** between them. Then λ is known, and $v = f\lambda$ gives the wave speed.

For a tube closed at one end and open at the other (a **resonance tube**), the closed end is a displacement node and the open end is a displacement antinode. The **fundamental** has $L = \lambda/4$; the next resonance is at $L = 3\lambda/4$; and so on. For a tube open at both ends, both ends are antinodes; the fundamental is at $L = \lambda/2$.

The general rule: a closed end is a node and an open end an antinode, and a string fixed at both ends has a node at each end. So a closed pipe fits an odd number of quarter-wavelengths ($L = \lambda/4, 3\lambda/4, 5\lambda/4, \ldots$), while an open pipe and a fixed string fit whole half-wavelengths ($L = \lambda/2, \lambda, 3\lambda/2, \ldots$); a pipe open at both ends with n nodes has $n + 1$ antinodes. Frequencies follow from $f = v/\lambda$: a closed pipe whose

lowest note is 820 Hz has $\lambda = 4L$, and a corridor 13.2 m long with a reflecting door at each end resonates lowest at $f = v/2L = 330/26.4 = 12.5$ Hz.

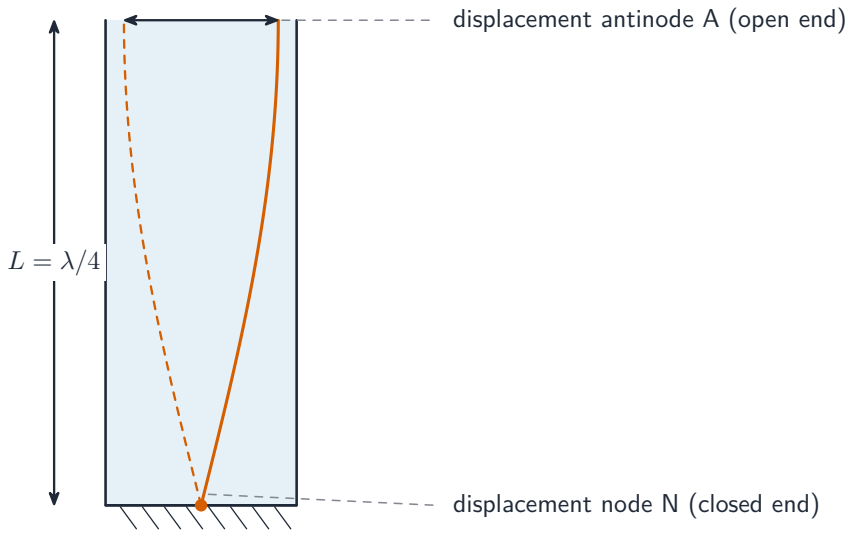


Modes of an air column: a closed end is a node, an open end an antinode

> **Worked example.** A loudspeaker at the open end of a tube 0.51 m long, closed at the other end, sets up a stationary wave with two nodes and two antinodes. Find the wavelength and, with $v = 340$ m s^{-1}, the frequency.

Two nodes and two antinodes is the second mode of a closed pipe, $L = 3\lambda/4$, so $\lambda = 4 \times 0.51/3 = 0.68$ m and $f = 340/0.68 = 500$ Hz. The longest wavelength that can resonate in this tube is $4L = 2.0$ m.

The experiments the syllabus names all measure λ from node or antinode spacing. In the **resonance tube**, a tube is raised out of water while a loudspeaker or tuning fork sounds at its open end; the sound is suddenly loud at the first resonance, when the air column is $\lambda/4$ long, and again at $3\lambda/4$, so the tube moves $\lambda/2$ between the two (a student needs no equipment to detect the resonance: the note becomes loud). In a **dust tube**, fine powder settles into heaps at the displacement nodes, $\lambda/2$ apart. With **microwaves**, a receiver moved along the line between the transmitter and a metal reflecting plate reads a minimum every $\lambda/2$: a receiver that starts at a minimum and passes six more minima in 1.05 m has crossed 3λ, so $\lambda = 0.35$ m and, with $v = 340$ m s^{-1} for the equivalent sound experiment, $f = 970$ Hz. A stationary wave in a microwave oven melts chocolate at the antinodes, $\lambda/2$ apart, so the spot spacing and the oven's frequency (2.45 GHz) give the speed of light from $c = f\lambda$.



Fundamental mode in a closed pipe – a node at the closed end, an antinode at the open end

Diffraction

Diffraction is the spreading of a wave after it passes through a gap or around an **obstacle**. All waves diffract – water, sound, light, microwaves.

For the two-mark "state what is meant by diffraction": **the spreading of a wave as it passes through a gap (an aperture) or around the edge of an obstacle**, into the region behind it. Diffraction changes the direction the wave travels in, but not its speed, frequency or wavelength.

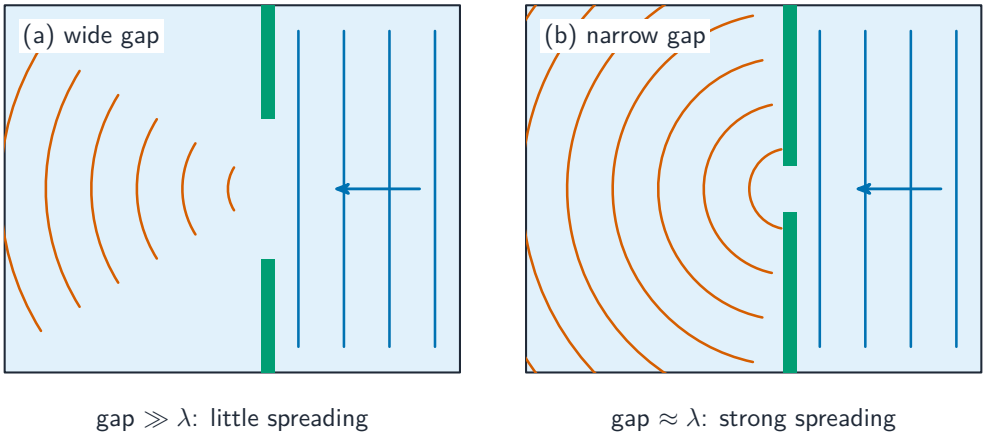
The amount of spreading depends on the ratio of **wavelength to gap width**:

- gap **much wider** than λ: very little spreading; the wave goes nearly straight through.
- gap **about the size of** λ: a lot of spreading; the wave fans out.
- gap **smaller than** λ: very strong spreading; the gap acts almost like a point source.

Show this with water waves in a **ripple tank**: straight waves meet a barrier with a gap, and the waves curve more as the gap is made narrower. The same idea is why you can hear someone around a corner (speech has λ near 1 m, close to the gap size) but cannot see them (visible light has $\lambda \sim 500$ nm, far smaller than the gap).

For the strongest spreading the gap should be about one wavelength wide. So to increase the diffraction of a given wave, make the gap narrower; for a given gap, use a longer wavelength, which means a lower frequency. Radio waves of wavelength 1.5 km diffract around a mountain and reach an aerial behind it; microwaves of 1.5 cm do not. Sound of 0.44 kHz in air has $\lambda = 330/440 = 0.75$ m, so features of about 0.75 m diffract it most, and of the sounds passing through a doorway 0.80 m wide the low

frequencies spread out most. Making the gap many wavelengths wide, or raising the frequency, reduces the spreading.



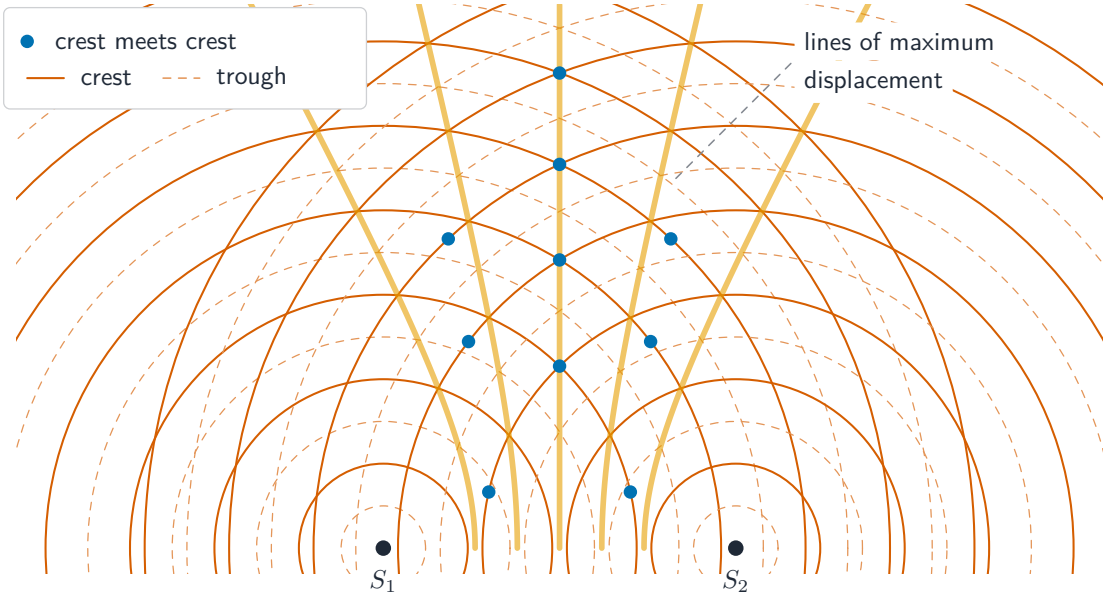
Diffraction in a ripple tank – a wide gap (a) spreads the waves little, a narrow gap (b) much more

Interference

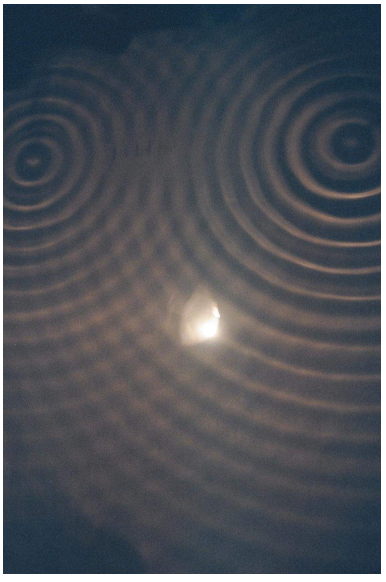


The shifting colours on a soap bubble come from the interference of light.

Interference is the superposition of two **coherent** waves to give a steady pattern of high-amplitude regions (constructive) and low-amplitude regions (destructive).



Two coherent sources give lines of maximum displacement where crests meet crests



The same effect in a real ripple tank – two coherent sources give a steady interference pattern

Coherence

Two sources are **coherent** when they emit waves with a **constant phase difference** (which also needs the same frequency). Two separate lamps are not coherent – their phase changes randomly, so any pattern flickers too fast to see and you get only an average.

The one-mark definition: coherent waves have a **constant phase difference**, which requires the same frequency. They need not be in phase with each other: two coherent sources emitting 180° apart give a pattern whose central line is a minimum. Two separate lasers, or a lamp and a laser, are not coherent even when their frequencies happen to match, so no steady pattern forms.

To make coherent light from one source, pass it through two **slits** in a **double-slit** setup. Both slits are lit by the same wavefront, so the two beams keep a fixed phase relationship.

Conditions for a clear pattern

To see two-source fringes you need:

1. two **coherent** sources (constant phase difference).
2. roughly equal amplitudes (or the dark regions are not very dark).
3. the waves overlap where you look.
4. for light (a transverse wave), the same plane of **polarisation** .

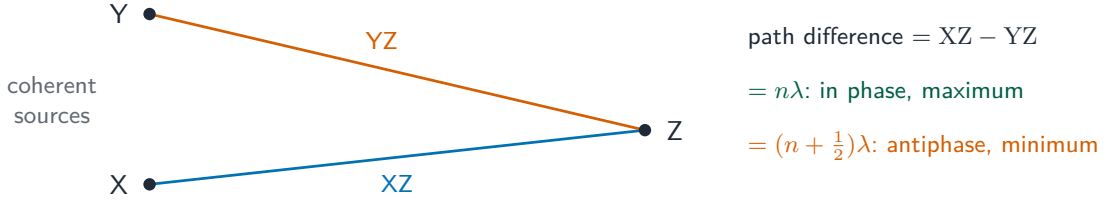
In practice, for light: a single slit (or a laser) makes the two slits coherent; the slits are narrow, so each diffracts the light into the region where the two beams overlap; and the slits are close together with the screen far away, so that the fringes are wide enough to see.

Path difference

For two coherent sources, what happens at a point depends on the **path difference** Δx between the two waves arriving there:

- constructive: $\Delta x = n\lambda$ (for whole numbers $n = 0, 1, 2, \ldots$).
- destructive: $\Delta x = (n + \frac{1}{2})\lambda$.

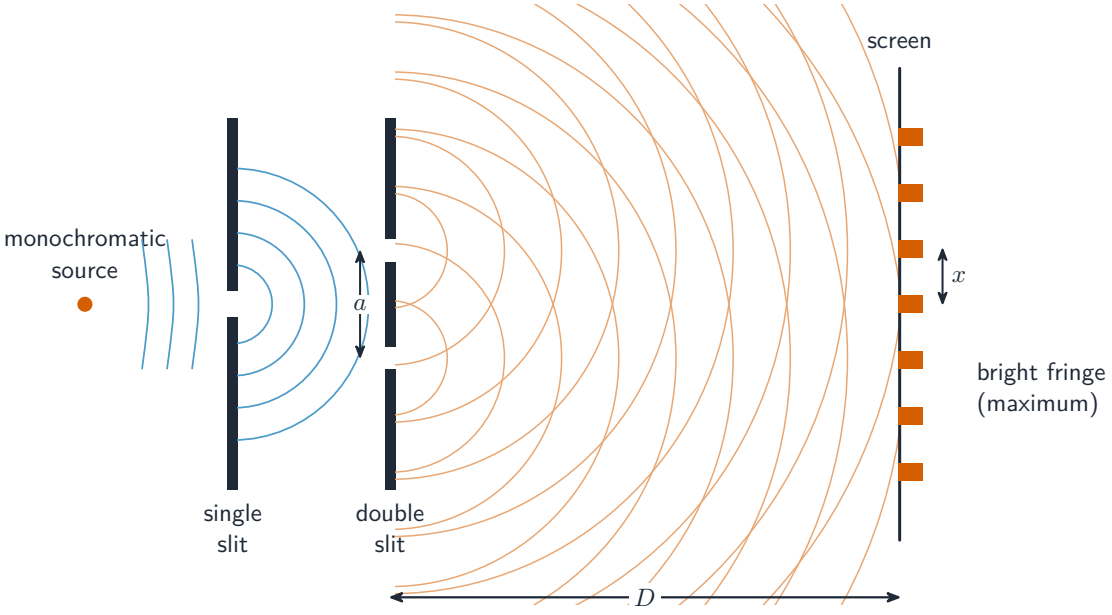
The path difference fixes the **phase difference**: one wavelength of path is 360°. Waves that have travelled 100 cm and 80 cm from two in-phase sources of wavelength 8.0 cm arrive with a path difference of 2.5λ, a phase difference of 180°, and cancel; microwaves of wavelength 4 cm whose paths differ by 6 cm (1.5λ) give a minimum, of zero intensity only if the two amplitudes are equal. Lowering both source frequencies equally lengthens the wavelength, so the same path difference is a smaller number of wavelengths and the point is no longer a minimum.



What happens at Z depends on the path difference XZ − YZ, measured in wavelengths

Double-slit (Young's) experiment

For two slits a distance a apart, with a screen a distance D away (assume $D \gg a$), light of wavelength λ makes fringes on the screen.



Young's double-slit experiment – the single slit makes the two slits coherent sources

The **fringe spacing** x (one **fringe** to the next) is

$$\lambda = \frac{ax}{D}, \qquad x = \frac{\lambda D}{a}.$$

Bright fringes (**maximum**) are where the path difference is a whole number of λ; dark fringes (**minimum**) where it is $(n+\frac{1}{2})\lambda$. The fringes are equally spaced.

To make the fringe spacing smaller: increase a (slits further apart), reduce D (screen closer), or use a shorter λ (bluer light).

Explain how the pattern of bright and dark fringes is formed (three marks): the light **diffracts** at each slit; the two diffracted beams **overlap** and superpose; where the path difference from the two slits is a whole number of wavelengths the waves arrive in phase and interfere constructively, giving a bright fringe, and where it is an odd number of half-wavelengths they arrive in antiphase and interfere destructively, giving a dark fringe. A brighter source makes the bright fringes brighter but does not change their spacing; changing to blue light makes the spacing smaller, so to keep the same spacing the slits must be moved closer together; making each slit narrower increases the diffraction, so fringes appear across a wider region, with the spacing unchanged.

> **Worked example.** In a double-slit experiment the slits are 0.50 mm apart and lit by light of wavelength 600 nm. The screen is 2.0 m away. Find the fringe spacing.

$$x = \frac{\lambda D}{a} = \frac{(600 \times 10^{-9})(2.0)}{0.50 \times 10^{-3}} = 2.4 \times 10^{-3} \text{ m} = 2.4 \text{ mm}.$$

> **Worked example.** Red light of wavelength 680 nm falls on slits 0.16 mm apart. The distance between the centres of the first and ninth dark fringes is 3.2 cm. Find the distance D to the screen.

Eight fringe spacings make 3.2 cm, so $x = 4.0$ mm and $D = ax/\lambda = (0.16\times10^{-3})(4.0\times10^{-3})/(680 \times 10^{-9}) = 0.94$ m. On a screen 5.0 cm wide centred on the pattern, with $x = 2.4$ mm, ten bright fringes fit on each side of the central one: 21 in all. A graph of x against a is a curve falling as $1/a$; a graph of x against λ (or against D) is a straight line through the origin with gradient D/a (or λ/a), from which a can be found.

Diffraction grating

A **diffraction grating** has many equally spaced slits – often hundreds or thousands per millimetre. Each slit is a coherent source. A maximum is seen at angle θ from the **normal** to the grating when

$$d \sin\theta = n\lambda,$$

where d is the **slit spacing** (centre to centre), $n = 0, \pm1, \pm2, \ldots$ is the **order**, and λ is the wavelength.

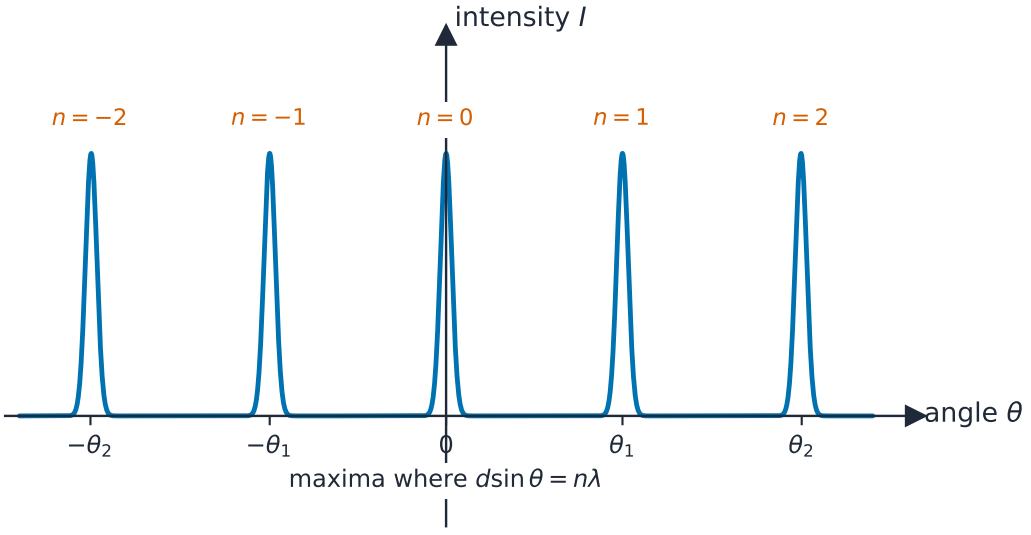
> **Worked example.** A diffraction grating has 500 lines per mm. Light of wavelength 600 nm is shone normally on it. Find the angle of the first-order ($n = 1$) maximum.

The slit spacing is $d = \dfrac{1}{500}$ mm $= 2.0 \times 10^{-6}$ m, so

$$\sin\theta = \frac{n\lambda}{d} = \frac{600 \times 10^{-9}}{2.0 \times 10^{-6}} = 0.30 \quad \Rightarrow \quad \theta \approx 17°.$$

Compared with the double slit, a grating gives much **sharper** maxima, because more slits add together – every other direction is cancelled by many slits.

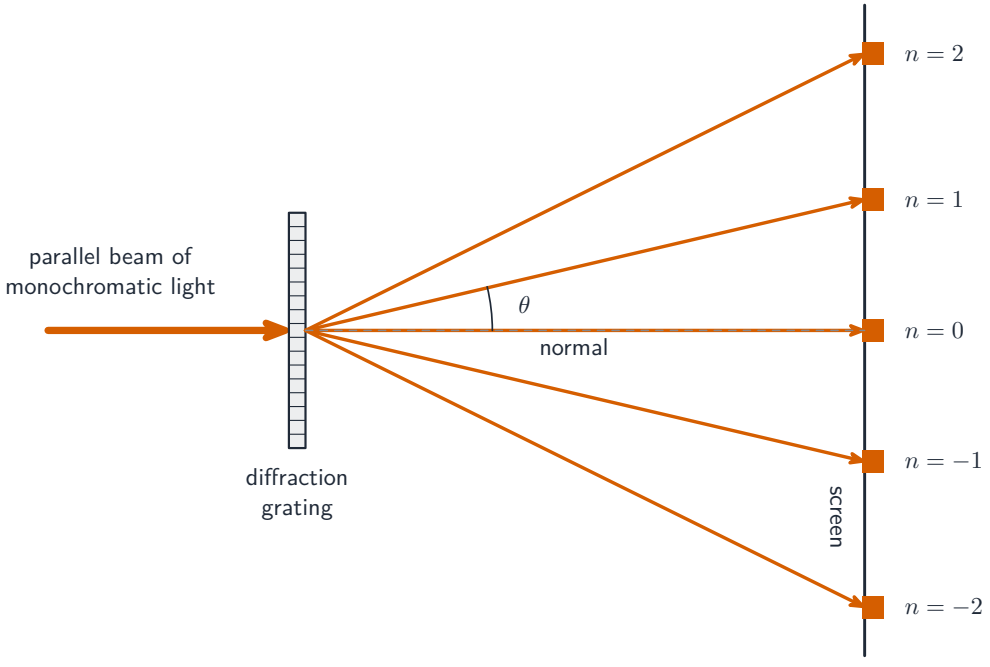
To **describe the diffraction at the grating**: the light spreads out (diffracts) at every slit, and the waves from all the slits superpose; in the directions where the path difference between neighbouring slits is a whole number of wavelengths they are all in phase, so sharp maxima form there and almost nothing in between. A graph of intensity against angle is a set of narrow peaks at $\theta = 0$ and at $\pm\theta_1, \pm\theta_2, \ldots$, where $\sin\theta_n = n\lambda/d$: equally spaced in $\sin\theta$, so slightly further apart in θ at the higher orders.



Intensity against angle for a grating: sharp maxima where $d\sin\theta = n\lambda$

Worked example. Light of wavelength 680 nm is incident normally on a grating with 450 lines per mm. Find the angle between the two second-order maxima.

$d = 1/450$ mm $= 2.22 \times 10^{-6}$ m, so $\sin\theta_2 = 2 \times 680 \times 10^{-9}/(2.22 \times 10^{-6}) = 0.612$ and $\theta_2 = 37.7°$; the two second-order beams are $2\theta_2 = 75°$ apart.



A diffraction grating splits monochromatic light into sharp maxima on a screen

■ Slit spacing from "lines per mm"

If a grating has N lines per millimetre, then $d = 1/N$ millimetres $= 10^{-3}/N$ metres. For 450 lines per mm, $d = 1/450$ mm ≈ 2.22 μm.

■ Highest order

For a given grating and wavelength, $\sin\theta = n\lambda/d$ cannot be more than 1, so the highest order seen is

$$n_{\max} = \left\lfloor \frac{d}{\lambda} \right\rfloor .$$

If $d/\lambda = 3.27$, orders up to $n = 3$ exist; $n = 4$ would need $\sin\theta > 1$ and is not seen.

So the total number of maxima on a wide screen is $2n_{\max} + 1$: for 700 nm light and 400 lines per mm, $d/\lambda = 3.57$, so $n_{\max} = 3$ and seven beams are seen. A shorter wavelength gives more orders at smaller angles. With white light every order except the zero order is a spectrum, violet nearest the centre and red furthest out, because θ grows with λ; the zero order stays white. Two wavelengths give a maximum at the same angle when $n_1\lambda_1 = n_2\lambda_2$: the third order of 400 nm coincides with the second order of 600 nm.

■ Finding λ with a grating

Shine parallel light of unknown wavelength straight at the grating. Measure the angle θ_1 of the first-order maximum from the centre. Then $\lambda = d\sin\theta_1$. Repeating for higher orders and averaging reduces error.

Measure the angle between the first-order maxima on the two sides and halve it, which cancels any error in setting the zero; higher orders give larger angles and so a smaller percentage uncertainty; and plotting $\sin\theta$ against n for several orders gives a straight line through the origin of gradient λ/d, so $\lambda = Gd$ (or, for a known wavelength, $d = \lambda/G$). Two things must be right: θ is measured from the **normal** to the grating, not from its surface, and d is the distance between adjacent lines, so 400 lines per mm means $d = 2.5$ μm, never 400.

■ source phase and path phase

Coherent sources have a **fixed** phase difference; they need not emit in phase. The phase difference at an observation point combines the initial source phase with the phase change due to the path difference. Use a consistent sign convention and reduce the result modulo 360°.

For a lagging second source, $\delta_{\text{arrival}} = \delta_{\text{source}} + 360°(r_2 - r_1)/\lambda$. A multiple of 360° gives a maximum; 180° modulo 360° gives a minimum (zero only for equal amplitudes).

> **Worked example.** Sources emit 180° apart. An extra path of $\lambda/2$ adds 180°, giving 360° at arrival: a **maximum**, not a minimum. In a wavefront diagram, a crossing of two crests is a maximum; a crest meeting a trough is a minimum.

■ Overlapping orders and a screen measurement

Two wavelengths give maxima in the same direction when $n_1\lambda_1 = n_2\lambda_2$. For example, second-order 600 nm and third-order 400 nm overlap because $n_1\lambda_1 = 2\times600 = 1200$ nm and $n_2\lambda_2 = 3 \times 400 = 1200$ nm.

If a grating order is measured on a screen a distance L along the normal, its displacement y gives $\theta = \tan^{-1}(y/L)$. Then apply $d\sin\theta = n\lambda$. This geometry differs from the small-angle double-slit fringe-spacing relation.

■ Demonstrating sound interference

Drive two loudspeakers from the same signal generator so their waves have the same frequency and a fixed source phase difference. Move a microphone, connected to an oscilloscope or sound-level display, across their overlapping sound field. Alternating large and small signal amplitudes show constructive and destructive interference. At a large-amplitude position the arrival phases match; at a small-amplitude position they oppose. Changing the microphone position changes the path difference. Unrelated speakers at the same nominal frequency need not maintain coherence.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
principle of superposition	when two or more waves meet at a point, the resultant displacement is the sum of the displacements of the individual waves
stationary wave	the pattern formed when two progressive waves of the same frequency and speed travel in opposite directions and superpose, with nodes and antinodes that do not move
node	a point on a stationary wave where the displacement is always zero
antinode	a point on a stationary wave where the amplitude is a maximum
diffraction	the spreading of a wave as it passes through a gap or around the edge of an obstacle
interference	the superposition of waves from coherent sources, giving a steady pattern of maxima and minima
coherence	waves that have a constant phase difference (and so the same frequency)

Term	Definition
path difference	the difference between the distances travelled by two waves from their sources to a point
fringe spacing	the distance between the centres of two adjacent bright (or dark) fringes
order of a maximum	the whole number n in $d\sin\theta = n\lambda$, the number of wavelengths of path difference between adjacent slits

Exam tips

- Two-source interference: **constructive** when path difference $= n\lambda$, **destructive** when $= (n + \frac{1}{2})\lambda$; the sources must be **coherent**.
- Double slit: $\lambda = ax/D$; diffraction grating: $d\sin\theta = n\lambda$ – know every symbol.
- On a **stationary wave** mark nodes and antinodes; adjacent nodes are $\lambda/2$ apart; it stores energy but does not transfer it.
- A stationary wave needs two waves of the **same frequency travelling in opposite directions**.

Common mistakes

- Adding amplitudes or intensities in the principle of superposition. Displacements add; the intensity then follows from the resultant amplitude squared.
- "Adjacent nodes are one wavelength apart." Half a wavelength; a node to the next antinode is a quarter.
- Using $d =$ lines per millimetre in $d\sin\theta = n\lambda$. Invert: $d = 10^{-3}/N$ metres.
- Measuring θ from the grating surface, or forgetting that the angle between the two first-order beams is $2\theta_1$.
- Writing "in phase" for coherent. Coherent means a constant phase difference; the sources may be permanently out of step.
- Saying the fringe spacing changes when the source is made brighter, or that narrower slits change the spacing. Brightness and slit width change the contrast and the number of visible fringes, not $x = \lambda D/a$.

Electricity

A-Level Physics

Electric current

An **electric current** is a flow of **charge carriers**. In a metal the carriers are negative conduction **electrons**; in an **electrolyte** they are positive and negative **ions**; in a **semiconductor** they may be electrons or "**holes**". The **conventional current** direction is the way **positive** charge would flow – opposite to the real flow of electrons in a wire.

Charge

Charge is **quantised**: the smallest free unit of charge is the **elementary charge**

$$e = 1.60 \times 10^{-19}\ \text{C}.$$

Every free charge in this syllabus is a whole-number multiple of e. The unit of charge is the **coulomb**, C.

So a particle can carry 4.8×10^{-19} C (three electrons' worth) or -2.4×10^{-18} C (fifteen), but never 1.1×10^{-19} C or 6.4×10^{-20} C; a set of measurements that gives a charge carrier 2.5×10^{-19} C cannot be right, because that is not a multiple of e. For the one-mark definitions: an **electric current is a flow of charge carriers**; the **coulomb** is the charge that passes a point in one second when the current is one ampere (an ampere multiplied by a second); a **charge carrier** can be an electron, a proton, an ion or an α-particle, but never a neutron.

Current as the rate of flow of charge

If charge Q passes a point in time t, the current is

$$I = \frac{Q}{t}, \qquad Q = It.$$

Unit of current: ampere, A ($= \text{C s}^{-1}$). For a changing current, the charge that has flowed in a time is the **area under** an I–t graph.

> **Worked example.** A current of 0.50 A flows for 2.0 minutes. Find the charge that passes, and how many electrons this represents. ($e = 1.60 \times 10^{-19}$ C.)

$$Q = It = 0.50 \times 120 = 60\ \text{C}, \qquad N = \frac{Q}{e} = \frac{60}{1.60 \times 10^{-19}} \approx 3.8 \times 10^{20}.$$

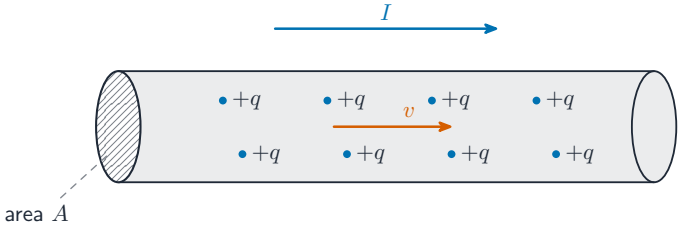
The same two lines answer "how many electrons pass in 10 hours at 4.0 mA" (convert the time to seconds: $N = It/e = 4.0 \times 10^{-3} \times 36\,000/(1.60 \times 10^{-19}) = 9.0 \times 10^{20}$) and,

the other way round, "the average current when 6.0×10^{23} electrons pass in 24 hours" ($I = Ne/t = 1.1$ A). A beam of α-particles carries a current too: each carries $2e$, so a beam current of 6.9×10^{-9} A is $6.9 \times 10^{-9}/(2 \times 1.60 \times 10^{-19}) = 2.2 \times 10^{10}$ particles per second. A lightning strike that transfers 1×10^{20} electrons in 30 μs is a current of $Ne/t = 5.3 \times 10^{5}$ A.

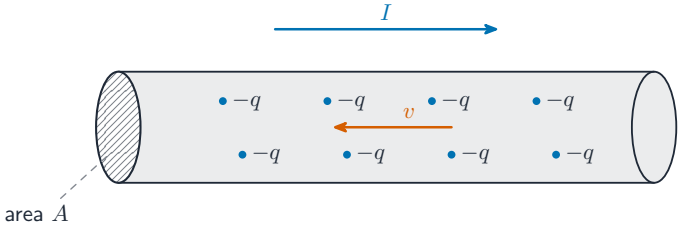
Drift velocity equation

For a uniform conductor of cross-section area A, with n charge carriers per unit volume (the **number density**), each carrying charge q, moving with average **drift velocity** v:

$$I = Anvq.$$



(a) positive carriers drift with I



(b) electrons drift against I

Charge carriers drifting inside a conductor: positive carriers drift with I, electrons against it

Worked example. A copper wire of cross-sectional area 1.0×10^{-6} m^2 carries a current of 5.0 A. Copper has $n = 8.5 \times 10^{28}$ free electrons per m^3. Find the drift velocity. ($e = 1.60 \times 10^{-19}$ C.)

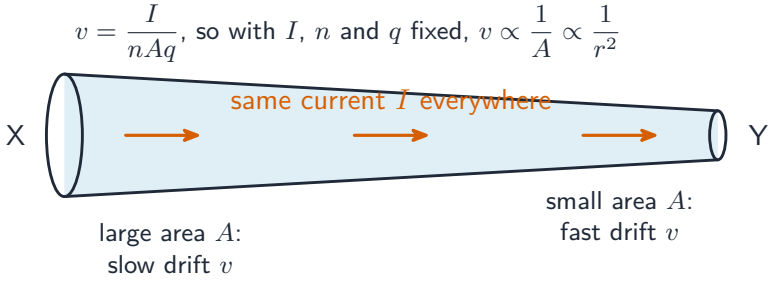
Rearranging $I = Anvq$ gives $v = \dfrac{I}{Anq}$:

$$v = \frac{5.0}{(1.0 \times 10^{-6})(8.5 \times 10^{28})(1.60 \times 10^{-19})} \approx 3.7 \times 10^{-4} \text{ m s}^{-1}.$$

The electrons drift very slowly – less than a millimetre per second.

Two "show that" steps that often precede this calculation. The **number density** of a metal with one free electron per atom is $n = \dfrac{\text{density} \times N_\text{A}}{\text{molar mass}}$: for copper, $8.9 \times 10^3 \times 6.02 \times 10^{23} / 0.0635 = 8.5 \times 10^{28}$ m^{-3}. And the **time** for an electron to drift the length of a wire is $t = L/v$: at 3.7×10^{-4} m s^{-1} a 2.0 m wire takes 5 400 s, an hour and a half, even though the current is established almost instantly. Between them, the equation $I = Anvq$ is used as a ratio far more often than as a calculation:

- **two wires in series** carry the same current; if wire Y has twice the diameter of X (four times the area) and the same metal, its electrons drift at a quarter of the speed. Two wires of the same length and diameter but different metals have drift speeds in the inverse ratio of their number densities.
- **a wire that narrows** (a wedge, a tapered rod, a cable of thick and thin strands) carries the same current at every cross-section, so the drift speed rises where the area falls: $v \propto 1/A \propto 1/r^2$. A sketch of v against distance along a tapering wire is a curve that rises more and more steeply towards the narrow end, not a straight line.
- the p.d. across a uniform wire is proportional to its length at fixed current, because $V = IR = I\rho L/A$ with I, ρ and A constant.



The same current everywhere, so the electrons drift faster where the wire is thinner

Use this to compare currents:

- a thinner wire (smaller A) at the same I needs a faster drift v.
- a semiconductor has far fewer free carriers than a metal (smaller n), so for the same I the drift velocity is much larger.
- in **series** components, I is the same everywhere, so if A stays the same but the material changes, nv changes the other way.

Potential difference

The **potential difference** (p.d.) across a component is the **energy** transferred per unit charge as that charge passes through it:

$$V = \frac{W}{Q}.$$

Unit: **volt**, V ($= \text{J C}^{-1}$).

If 1 J of electrical energy changes into other forms (thermal, light, kinetic, …) when 1 C of charge passes through a component, the p.d. across it is 1 V.

The **electromotive force** (e.m.f.) of a source is the energy given **per unit charge by the source**. The formula is the same as for p.d.; the difference is direction: e.m.f. is energy given **to** the charge by the change; p.d. is energy given **up** by the charge to the component.

The definitions the examiner accepts: the **potential difference** across a component is the energy transferred (from electrical to other forms) per unit charge passing through it; the **e.m.f.** of a source is the energy transferred from other forms (chemical, in a cell) to electrical energy per unit charge in driving the charge round a complete circuit; one **volt** is one joule per coulomb. A battery "marked 9.0 V" therefore gives each coulomb 9.0 J of electrical energy for the whole circuit, and the product of charge and p.d. is the energy transferred.

Electrical power



High-voltage power lines carry electrical energy across the country.

Combining $V = W/Q$ and $I = Q/t$:

$$P = \frac{W}{t} = VI.$$

Using Ohm's law $V = IR$:

$$P = VI = I^2R = \frac{V^2}{R}.$$

Pick the form with the quantities you know. Examples:

- two heaters of equal resistance – the one with the larger current gives more **power** ($P = I^2R$).
- two resistors in **parallel** across the same **voltage** – the one with smaller R gives more power ($P = V^2/R$).
- a kettle marked "2.4 kW, 240 V" draws $I = P/V = 10$ A and has resistance $R = V^2/P = 24\ \Omega$.

Energy transferred in time t is $E = Pt$.

> **Worked example.** Two lamps, P rated 250 V, 50 W and Q rated 250 V, 200 W, are connected in series to a 250 V supply. Which is brighter?

Their resistances at the rated p.d. are $R = V^2/P$: 1250 Ω for P and 313 Ω for Q. In series they carry the same current, so $P = I^2R$ makes the larger resistance, lamp P, dissipate more power: the "weaker" lamp is the brighter one. In **parallel** the p.d. is the same across both and $P = V^2/R$ favours the smaller resistance: Q. "State and explain which resistor dissipates more power" is answered by naming the quantity the two share (current in series, p.d. in parallel) and the form of the power equation that uses it.

> **Worked example.** A supply delivers 2.4 kW at 240 V to a kettle through two cables of resistance 0.10 Ω each. Find the power lost in the cables.

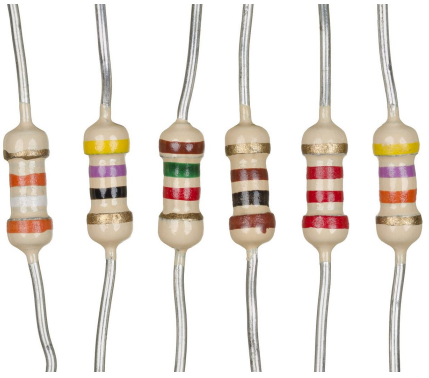
The current is $I = P/V = 10$ A, so the cables dissipate $I^2R = 10^2 \times 0.20 = 20$ W, and the kettle receives 2380 W. The **efficiency** of a circuit that exists to power one component is that component's power divided by the total power from the supply: with 6.0 V across a 0.86 Ω series resistor and 4.5 V across the component at the same current, the efficiency is $4.5/(6.0 + 4.5) = 43\%$. A thermistor across a fixed p.d. dissipates $P = V^2/R$, so as its temperature rises and R falls, the power rises. A kettle draws about 10 A from a 250 V supply; a mobile phone charger a fraction of an ampere.

Resistance and Ohm's law

The **resistance** R of a component is

$$R = \frac{V}{I}.$$

Unit: **ohm**, Ω ($= \text{V A}^{-1}$). Resistance depends on the conditions (such as temperature) when it is measured.



Real fixed resistors – the coloured bands code the resistance in ohms

■ Ohm's law

A conductor obeys **Ohm's law** when the current through it is **proportional to the p.d.** across it, as long as the conditions (especially temperature) stay constant. For such a conductor R is constant and the I–V graph is a straight line through the origin.

Ohm's law is an experimental result, not a definition. The definition $R = V/I$ works for **any** component; only ohmic ones have constant R.

For the marks: **resistance** is the ratio of the potential difference across a component to the current in it; the **ohm** is the resistance of a component in which a p.d. of one volt produces a current of one ampere (a volt per ampere); **Ohm's law** states that the current in a metallic conductor is directly proportional to the potential difference across it, provided that its temperature (and other physical conditions) remains constant. Of several I–V graphs, only a straight line through the origin obeys Ohm's law; a straight line that misses the origin, or any curve, does not.

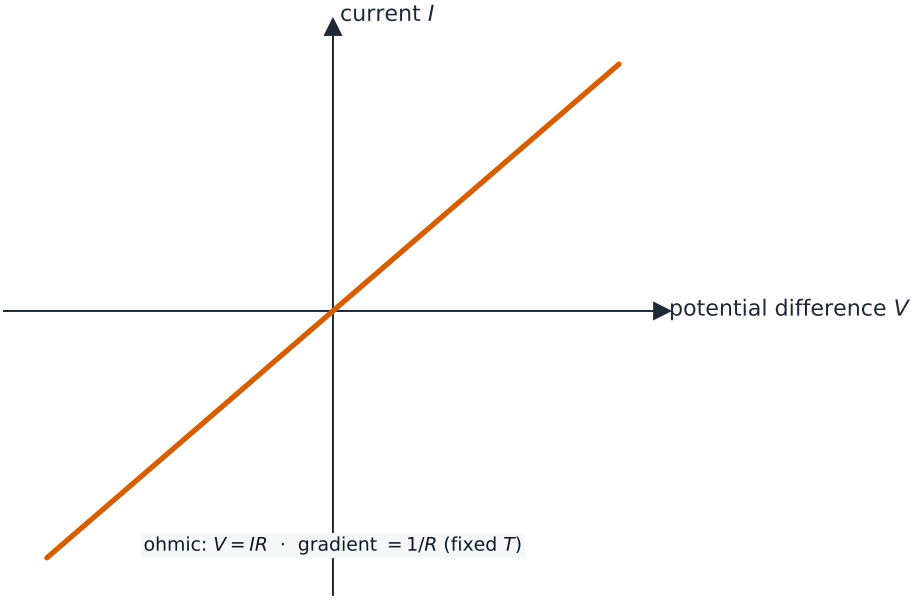
■ I–V characteristics

You should be able to sketch these:

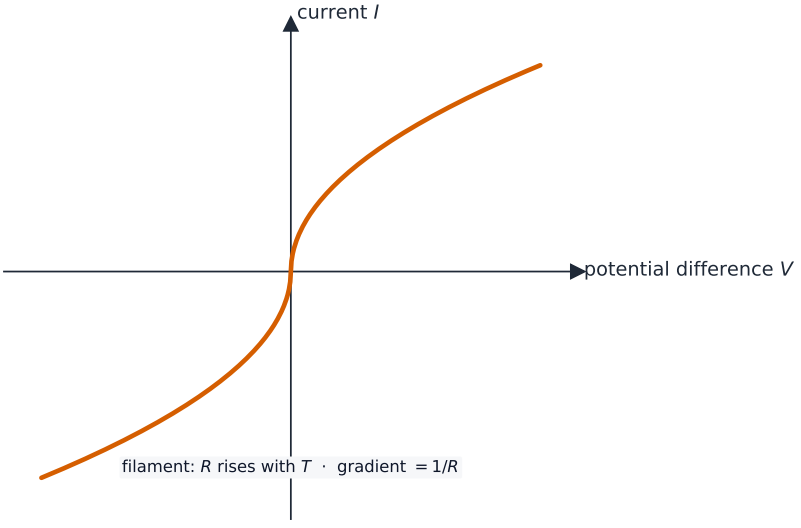
- **metal wire at constant temperature** – a straight line through the origin (constant R). Reversing the p.d. drives the current the other way, giving a straight line in both directions.
- **filament lamp** – through the origin, steep at first, then flatter as V (and I) grow. Reason: more current heats the filament, so its resistance rises and the gradient $1/R$ falls.
- **semiconductor diode** – almost no current for negative V or small positive V. Above a "switch-on" voltage (about 0.7 V for silicon), the current rises sharply.

Explain the shape of the filament lamp's line (three marks): as the current increases the filament's temperature rises; the lattice ions vibrate with larger amplitude, so the free electrons collide with them more often; each collision takes energy from the electrons, so the resistance increases; on the graph the ratio V/I grows and the gradient falls. Run backwards for "the current decreases": the temperature falls,

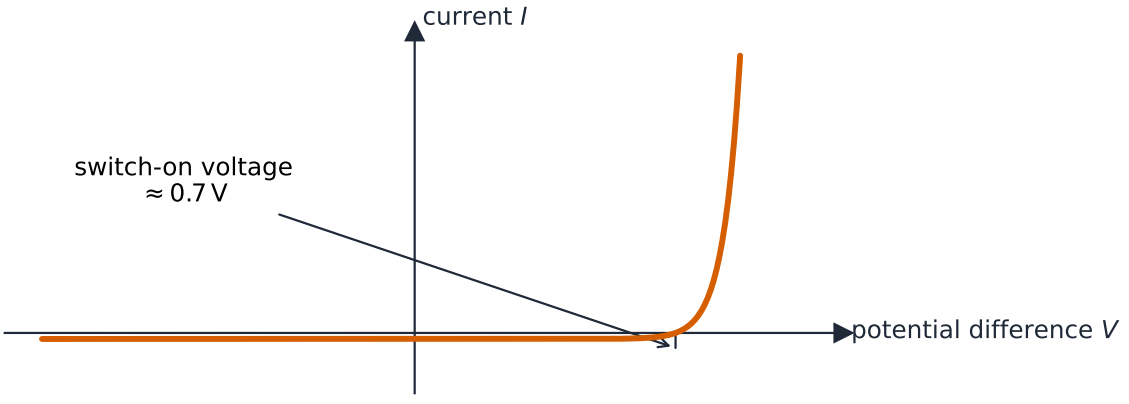
so the resistance falls. Two things to read off any characteristic. The resistance at a point is V/I for that point, never the gradient of a curve, so a diode's resistance is very large (infinite, in practice) up to the switch-on p.d. and then falls steeply as V rises further, and of four components at the same p.d. the one with the smallest current has the greatest resistance. And when a lamp (or a diode) is in series with a resistor, the two carry the same current and their p.d.s add: read the current from the component's curve at its own p.d., then use $V = IR$ for the resistor, and the supply p.d. is the sum.



I–V characteristic of an ohmic conductor (metal wire at constant temperature)



I–V characteristic of a filament lamp



I–V characteristic of a semiconductor diode

Resistivity

For a uniform conductor of length L and cross-section area A,

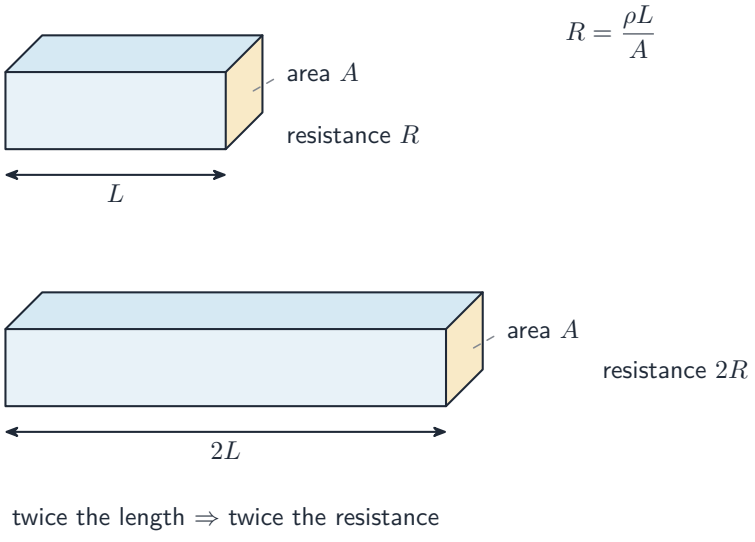
$$R = \frac{\rho L}{A}.$$

ρ is the **resistivity** , a property of the material, with unit Ω m. Doubling the length doubles R; doubling the area halves it; halving the diameter quarters the area and so makes R four times bigger.

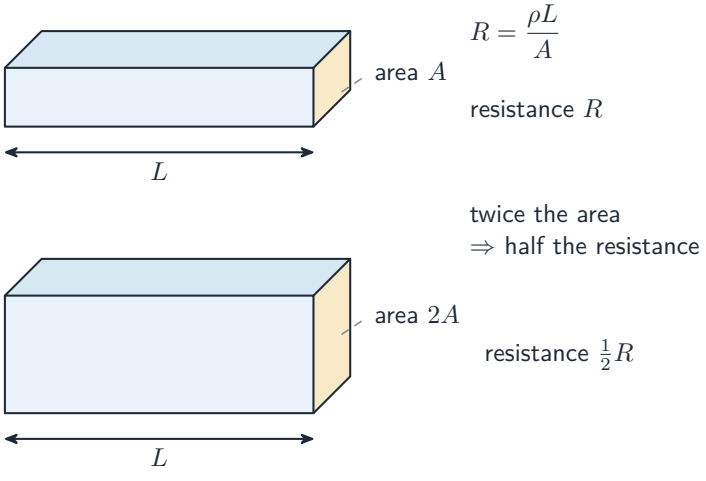
> **Worked example.** A copper wire of length 2.0 m and cross-sectional area 1.7×10^{-7} m^2 has resistivity 1.7×10^{-8} Ω m. Find its resistance.

$$R = \frac{\rho L}{A} = \frac{(1.7 \times 10^{-8})(2.0)}{1.7 \times 10^{-7}} = 0.20\ \Omega.$$

Typical values: copper at room temperature $\rho \sim 1.7 \times 10^{-8}$ Ω m; an **insulator** $\rho \sim 10^{15}$ Ω m or more.



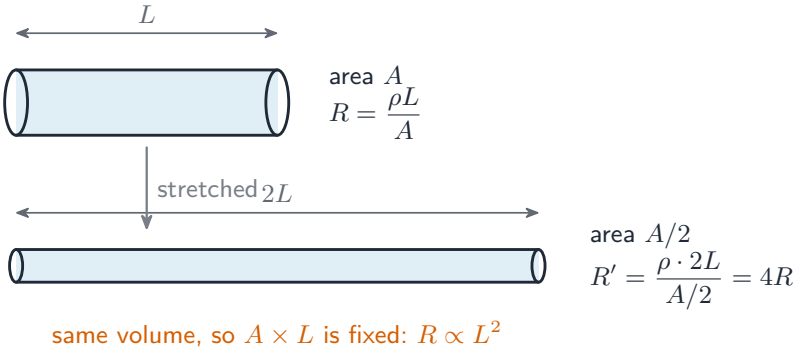
A longer conductor has more resistance – doubling L doubles R



A wider conductor has less resistance – doubling A halves R

The resistivity of a metal **rises with temperature** (more **lattice vibration** **scatters** the electrons), which is why the filament lamp's I–V line curves.

Most resistivity questions are ratios. **Stretching** a wire keeps its volume ($A \times L$) constant, so if the length becomes k times longer the area becomes k times smaller and $R = \rho L/A$ becomes k^2 times larger: a wire three times as long (same mass, same metal) has nine times the resistance, and a wire whose diameter falls to 0.940 of its value has its area multiplied by 0.884, its length divided by 0.884, and its resistance multiplied by $1/0.884^2 = 1.28$.



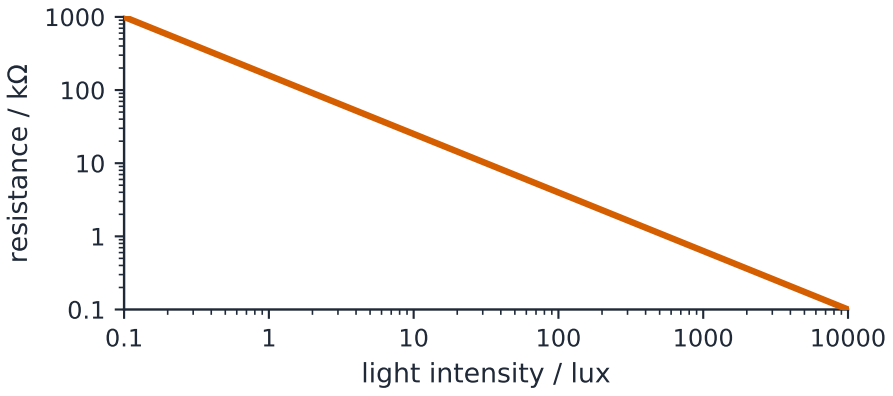
Stretched at constant volume: twice the length, half the area, four times the resistance

> **Worked example.** A copper lightning rod of resistance 9.6 Ω and length 20 m has resistivity 1.7×10^{-8} Ω m. Find its radius.

$A = \rho L/R = 1.7 \times 10^{-8} \times 20/9.6 = 3.5 \times 10^{-8}$ m^2, and $r = \sqrt{A/\pi} = 1.1 \times 10^{-4}$ m. Doubling the radius (same length) quarters the resistance. The other standard ratios: **strands in parallel** – seven identical strands share the current, so the cable's resistance is one seventh of one strand's; two wires of the same resistance where one metal has twice the resistivity need the second wire to have twice the area (a diameter $\sqrt{2}$ times larger) for the same length; a wire's resistance per unit length, 0.92 Ω m^{-1}, multiplied by its area, 5.3×10^{-7} m^2, is its resistivity, 4.9×10^{-7} Ω m; and a cylinder of conducting putty 60 mm long and 20 mm across, or a cube of side a ($R = \rho a/a^2 = \rho/a$), uses the same $R = \rho L/A$ with the shape's own length and end area. When the wire is held under tension, $R_0 = \rho L/A$ still gives its resistance, and a stretch that lengthens it and thins it raises R for both reasons. Because $\rho = RA/L = R\pi d^2/(4L)$, the percentage uncertainty in a measured resistivity is the sum of the percentage uncertainties in R and L plus **twice** that in d.

Light-dependent resistor (LDR)

A **light-dependent resistor** (LDR) is a semiconductor whose **resistance falls as the light intensity rises**. In bright light R may be a few hundred Ω; in the dark it can be in the megaohms. LDRs are used in light-sensing circuits (street lamps, camera light meters). Here the **light intensity** controls the resistance.

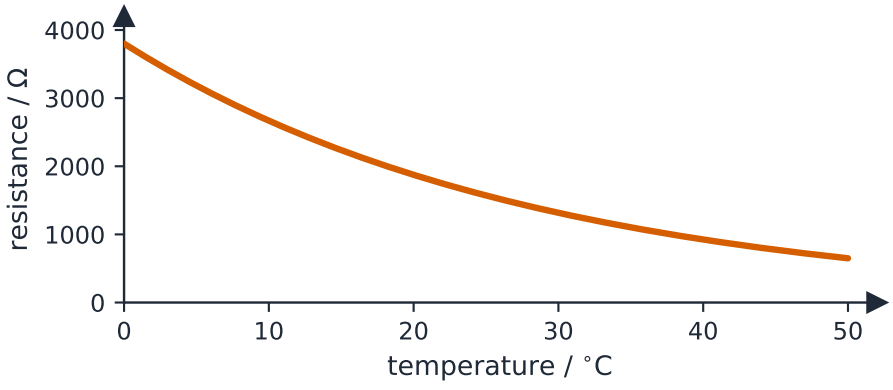


LDR: more light ⇒ lower R

Resistance of an LDR decreases as light intensity increases

Thermistor

In this syllabus a **thermistor** has a **negative temperature coefficient** : its **resistance falls as its temperature rises**. This is useful for sensing temperature – put it in a **potential divider** and the output voltage changes with temperature.



NTC: R falls as T rises

Resistance of a thermistor falls as temperature rises

This is the opposite of a metal: in a semiconductor, more thermal energy frees more charge carriers, and this matters more than the extra scattering.

A sketch of a thermistor's resistance against temperature starts at R_0 at 0 °C and falls along a curve that flattens: the fall is not linear, which is the disadvantage of a

thermistor as a thermometer (its scale is not uniform, so it needs calibrating). When the light on an LDR is increased, or a thermistor is warmed, its resistance falls, the current in its circuit rises, and the p.d. across it (in series with a fixed resistor) falls while the p.d. across the resistor rises; a fixed resistor and a metal wire at constant temperature keep their resistance, and a filament lamp's rises with current.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
electric current	a flow of charge carriers
coulomb	the charge passing a point in one second when the current is one ampere
potential difference	the energy transferred from electrical to other forms per unit charge passing through a component
electromotive force (e.m.f.)	the energy transferred from other forms to electrical energy per unit charge, by a source driving charge round a complete circuit
volt	one joule per coulomb
resistance	the ratio of the potential difference across a component to the current in it
ohm	the resistance of a component in which a potential difference of one volt produces a current of one ampere
Ohm's law	the current in a metallic conductor is directly proportional to the potential difference across it, provided its temperature remains constant
resistivity	the constant ρ in $R = \rho L/A$, a property of the material with unit Ω m
number density	the number of charge carriers per unit volume of the conductor

Exam tips

- Use $I = Q/t$, $V = W/Q$ (energy per unit charge) and $P = VI = I^2R = V^2/R$.
- **Ohm's law ($V = IR$)** applies only to an ohmic conductor at constant temperature – a filament lamp is non-ohmic.
- Sketch and interpret the **I-V characteristics** of a resistor, filament lamp and diode.
- An **LDR's** resistance falls with light; a **thermistor's** falls as temperature rises.

Common mistakes

- Taking the resistance from the gradient of a curved I–V graph. Resistance is V/I at the point; only for a straight line through the origin is it the reciprocal of the gradient.
- Leaving a time in minutes or hours in $Q = It$. Convert to seconds first.
- Saying the filament lamp's resistance rises "because of the voltage". The chain is current, temperature, ion vibration, more collisions, more resistance.
- Forgetting that a stretched wire changes in two ways. At constant volume the area falls as the length rises, so $R \propto L^2$.
- Using the diameter as the radius in $A = \pi r^2$, or leaving mm^2 unconverted.
- Comparing powers with the wrong form. Series components share the current, so use I^2R; parallel components share the p.d., so use V^2/R.

D.C. circuits

A-Level Physics

Practical circuits

e.m.f. and p.d.

The **electromotive force** (e.m.f.) ε of a source is the **energy** given to each unit of charge by the source as it drives the charge around a full circuit. Unit: volt.

The **potential difference** (p.d.) across a component is the energy changed from electrical to other forms by each unit of charge as it passes through that component.

Both are in volts; they differ in energy direction:

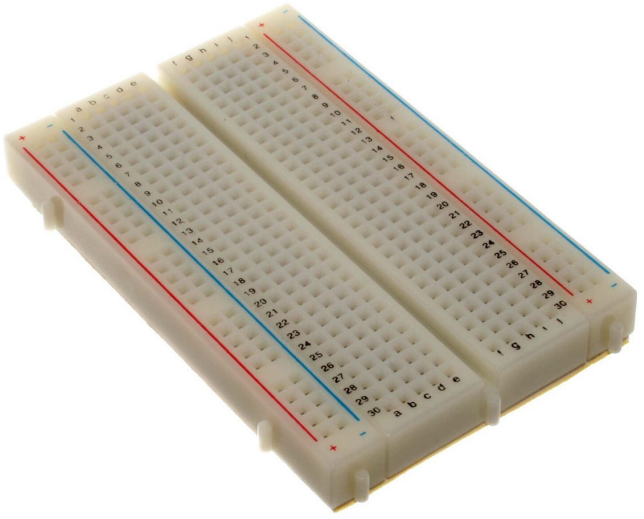
- e.m.f. – energy put **into** the circuit by the source (chemical → electrical in a battery, mechanical → electrical in a generator).
- p.d. – energy taken **out** of the electrical form (electrical → thermal in a resistor, → light in a lamp, → kinetic in a motor).

The examiner's wording is fixed. e.m.f. is "the energy transferred **per unit charge** by the source in driving charge round a **complete circuit**"; p.d. is "the energy transferred per unit charge from electrical to other forms". Both are energy per charge, so the volt is a joule per coulomb ($1\ \text{V} = 1\ \text{J C}^{-1}$). Statements that are **always** true of a source: the e.m.f. is the terminal p.d. when **no current** flows; the total energy it gives to a charge Q is εQ. The name is a trap: an electromotive "force" is not a force and is not measured in newtons.

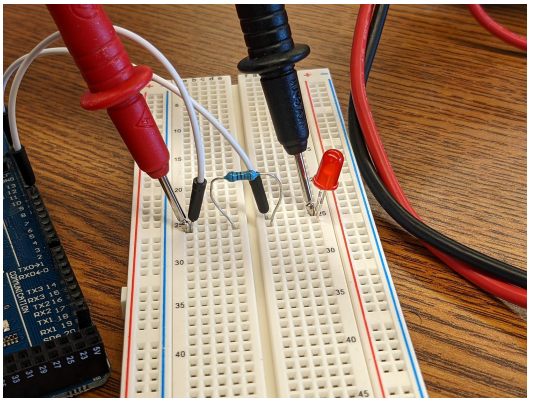
> **Worked example.** A cell of e.m.f. ε and internal resistance r drives a charge Q round a circuit whose terminal p.d. is V. Compare the energy transferred by the cell with the energy dissipated outside it.

The cell transfers εQ in total; the external circuit receives VQ. Since $V = \varepsilon - Ir < \varepsilon$ whenever a current flows, $VQ < \varepsilon Q$ – the difference $(\varepsilon - V)Q = IrQ$ is the energy turned to heat inside the cell.

In the lab you often build a circuit on a **breadboard** (a board with rows of holes that connect components without soldering) and measure currents and p.d.s with a **multimeter**.



The bare board: the two long rails carry power, and each short row of holes is joined underneath



A real circuit on a breadboard, being measured with a multimeter

Internal resistance

A real source has some **internal resistance** r – usually the resistance of the **electrolyte** in a **cell**, or the wire windings in a generator. When current I flows, an internal p.d. of Ir is "lost" inside the source, so the **terminal p.d.** across the outside circuit is

$$V_{\text{terminal}} = \varepsilon - Ir.$$

So:

- **no current** (**open circuit**, $I = 0$): the terminal p.d. equals the e.m.f.
- **larger current**: the terminal p.d. falls.

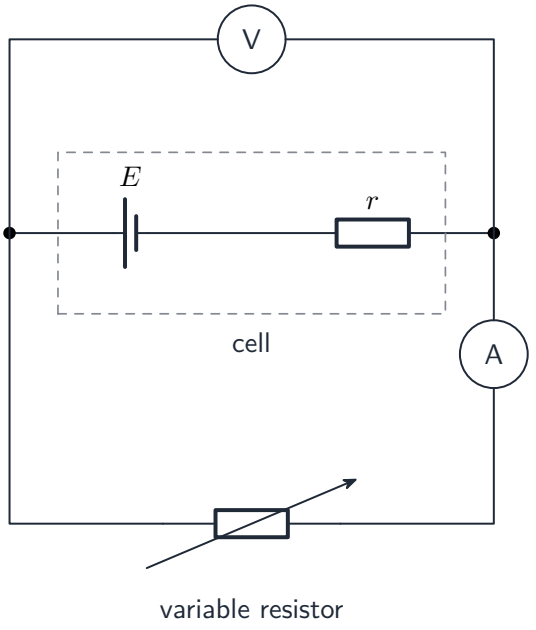
- **short circuit** ($R_{\text{external}} \to 0$): $I = \varepsilon/r$, a large current, with all the energy turned to heat inside the source.

To measure r, change the outside resistance and plot V_{terminal} against I: the line has y-intercept ε and gradient $-r$.

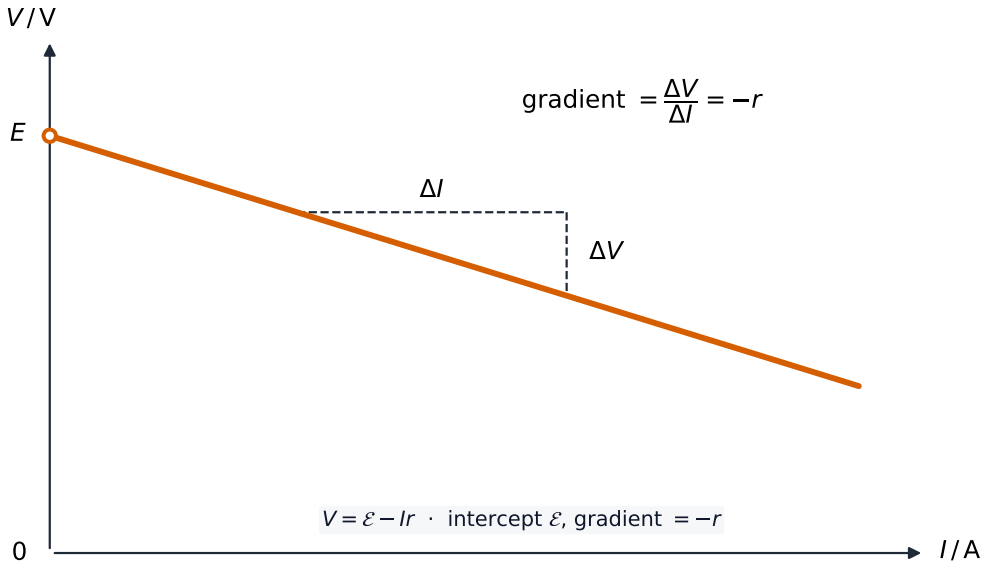
> **Worked example.** A cell of e.m.f. 1.5 V and internal resistance 0.50 Ω is connected to a 2.5 Ω resistor. Find the current and the terminal p.d.

The e.m.f. drives the current through both resistances: $I = \dfrac{\varepsilon}{R+r} = \dfrac{1.5}{2.5+0.50} = 0.50$ A. Then

$$V_{\text{terminal}} = \varepsilon - Ir = 1.5 - 0.50 \times 0.50 = 1.25 \text{ V}.$$



Circuit for measuring the e.m.f. and internal resistance of a cell



Terminal p.d. against current – the intercept is the e.m.f. and the gradient is minus the internal resistance

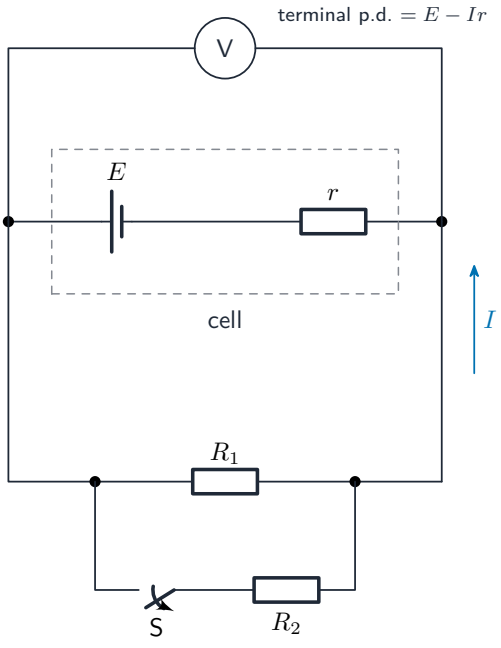
The **power** given to the outside load is $P_{\text{ext}} = (\varepsilon - Ir)I$; the power lost inside is $P_{\text{int}} = I^2 r$; the total power from the source is εI.

The **efficiency** of the source is $\dfrac{P_{\text{ext}}}{\varepsilon I} = \dfrac{VI}{\varepsilon I} = \dfrac{V}{\varepsilon} = \dfrac{R}{R+r}$: a large load resistance wastes little energy inside the source.

> **Worked example.** A cell of e.m.f. 2.0 V and internal resistance 0.40 Ω drives a current of 1.5 A through a wire. Show that the terminal p.d. is 1.4 V and find the percentage efficiency with which the cell supplies power to the wire.

$V = \varepsilon - Ir = 2.0 - 1.5 \times 0.40 = 2.0 - 0.60 = 1.40$ V. Power to the wire $= VI = 1.4 \times 1.5 = 2.1$ W; total power from the cell $= \varepsilon I = 2.0 \times 1.5 = 3.0$ W. Efficiency $= 2.1/3.0 = 0.70$, i.e. 70%. In a "show that" question, write every step and the unrounded value (1.40 V) before the value you were given.

When the outside circuit changes, argue through the current. Closing a switch that adds a second resistor **in parallel** lowers the total external resistance, so the current in the cell rises, the internal "lost volts" Ir rise, and the terminal p.d. $\varepsilon - Ir$ falls; the p.d. across the original resistor (which is the terminal p.d.) therefore falls, and so does its current, even though the cell's current rose. This chain – resistance → current in the cell → Ir → terminal p.d. – is the model answer for almost every "state and explain the effect" question in this topic.

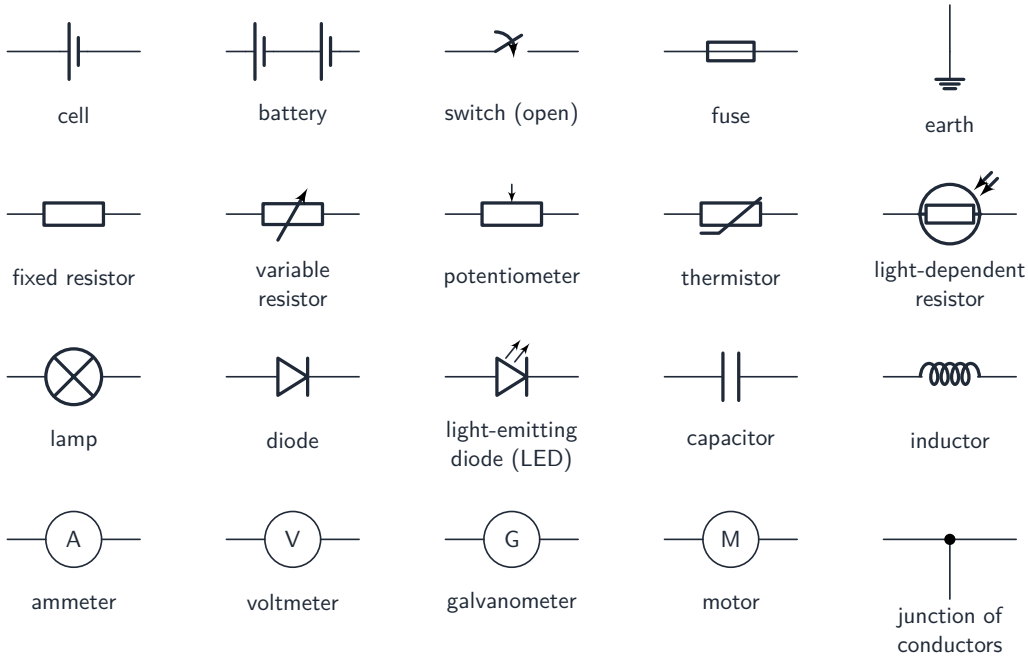


Closing S adds a parallel branch: the current in the cell rises, so Ir rises and the terminal p.d. read by the voltmeter falls

Cells joined together. In series, e.m.f.s add and internal resistances add: three cells of e.m.f. ε and internal resistance r give 3ε and $3r$. A cell connected the **wrong way round** subtracts its e.m.f. (ten 1.5 V cells with one reversed give $8 \times 1.5 = 12$ V). Identical cells in parallel give the same e.m.f. ε with a smaller internal resistance r/N, so they can supply a larger current.

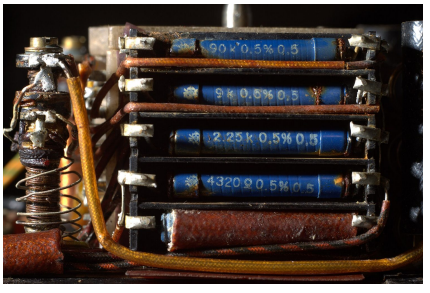
■ Circuit symbols

You must recognise and draw the standard symbols in the syllabus: cell, battery, switch, resistor, variable resistor, **ammeter** , **voltmeter** , lamp, **diode** (and **LED**), **capacitor** , inductor, thermistor, light-dependent resistor, **fuse** , earth, junction. An ideal ammeter has zero **resistance** and goes in **series** . An ideal voltmeter has infinite resistance and goes in **parallel** .

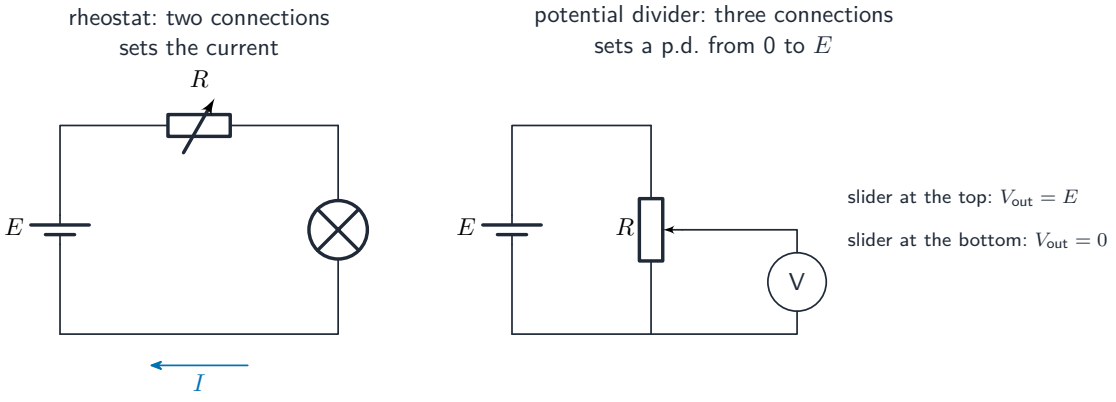


The standard circuit symbols you need to recognise and draw

Drawing a circuit diagram. Marks are lost for symbols, not physics. Use ruler-straight lines and the standard symbols; an ammeter goes **in series** with the component whose current it measures and a voltmeter **in parallel** with (across) the component whose p.d. it measures; a cell with internal resistance is drawn as an ideal cell in series with a resistor r; and a variable resistor may be used in two ways, shown below. A "complete the circuit diagram" question usually wants exactly this measuring arrangement: source, ammeter and variable resistor in one series loop, voltmeter across the component under test.



A digital multimeter across a resistor bank: the voltmeter goes in parallel, the ammeter in series



A variable resistor as a rheostat (two connections, sets the current) and as a potential divider (three connections, sets a p.d. from 0 to E)

■ Audio-device symbols



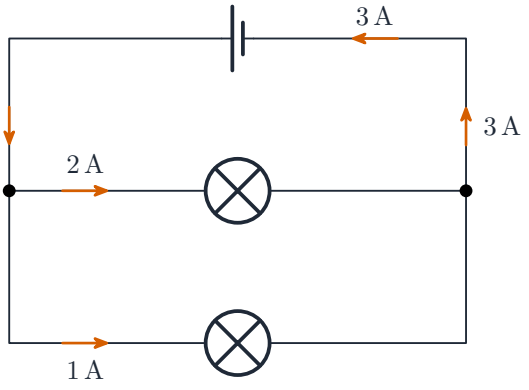
A **buzzer** and an **electric bell** turn electrical energy into sound; a **loudspeaker** turns a varying electrical signal into sound; a **microphone** turns sound into a varying electrical signal. Learn their distinct outlines and connections. The cup-shaped symbol with a flat top and two leads is the buzzer used in the AS examination; a device's function alone does not identify the symbol.

Kirchhoff's laws

■ First law (junction rule)

At any **junction** , **the total current flowing in equals the total current flowing out**. This follows from **conservation of charge** – charge cannot build up at a point in a steady circuit, so charge in per second equals charge out per second.

For a junction with three wires: $I_1 = I_2 + I_3$ if currents 2 and 3 flow out and current 1 flows in.



Current divides at a junction in a parallel circuit (3 A in equals 2 A plus 1 A)

Two ways the examiner asks it: "state the law" (one mark: the sum of the currents **into** a junction equals the sum of the currents **out of** it) and "state the conservation law behind it" (charge). Do not answer "energy" for the first law or "charge" for the second – the pairing is tested in almost every Paper 1.

■ Second law (loop rule)

Around any **closed loop** , **the total e.m.f. equals the total p.d. across the components in that loop**. This follows from **conservation of energy** : as a unit of charge goes once round a loop, the energy it gains from sources equals the energy it gives up to components.

Pick a direction round the loop. Take an e.m.f. as positive when the loop direction goes from − to + of the source, and a p.d. as positive when the loop direction is the conventional current direction through the resistor.

In symbols, round any closed loop $\sum \varepsilon = \sum IR$. A source you pass from + to − counts as a **negative** e.m.f. (it is being charged, or opposes the other source), and a resistor you pass **against** its current counts as a negative p.d.

> **Worked example.** Two batteries are in one loop with their e.m.f.s opposed: 12.0 V with internal resistance 1.0 Ω, and 8.0 V with internal resistance 0.50 Ω. Find the current.

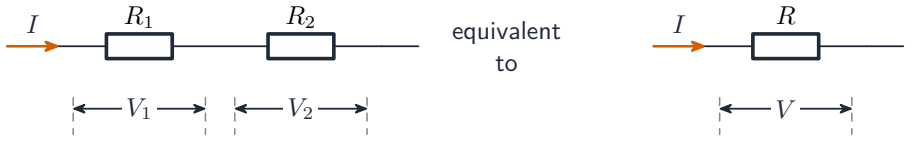
Going round the loop, the net e.m.f. is $12.0 - 8.0 = 4.0$ V and the total resistance is $1.0 + 0.50 = 1.5\ \Omega$ (the internal resistances are in series), so $I = 4.0/1.5 = 2.7$ A. The current flows in the direction driven by the larger e.m.f.

Combining resistors

Resistors in series. *Derivation using Kirchhoff's laws:* there is no junction between the resistors, so by the **first law** the same current I passes through each. By the **second law** the e.m.f. round the loop equals the sum of the p.d.s:

$$\varepsilon = IR_1 + IR_2 + \ldots = I(R_1 + R_2 + \ldots),$$

so $R_{\text{series}} = R_1 + R_2 + \ldots$.

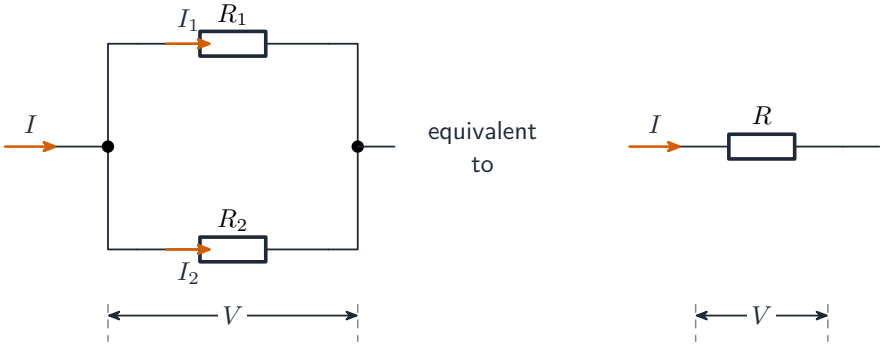


Two resistors in series and their single equivalent resistor

Resistors in parallel. *Derivation:* by the **second law**, each resistor forms its own loop with the source, so each has the same p.d. V across it. By the **first law** the current entering the junction equals the sum of the branch currents:

$$I = \frac{V}{R_1} + \frac{V}{R_2} + \ldots = V\left(\frac{1}{R_1} + \frac{1}{R_2} + \ldots\right),$$

so $\dfrac{1}{R_{\text{parallel}}} = \dfrac{1}{R_1} + \dfrac{1}{R_2} + \ldots$.



Two resistors in parallel and their single equivalent resistor

Two equal resistors R in parallel give $R/2$; N equal ones give R/N. A parallel combination is **always smaller** than any of its resistors; a series combination is always larger.

> **Worked example.** A 4.0 Ω resistor and a 12 Ω resistor are connected in parallel. Find their combined resistance.

$$\frac{1}{R} = \frac{1}{4.0} + \frac{1}{12} = \frac{3}{12} + \frac{1}{12} = \frac{4}{12} = \frac{1}{3} \quad \Rightarrow \quad R = 3.0\ \Omega.$$

Resistor networks and power

Reduce a mixed network **one step at a time**: replace each series chain by its sum and each parallel pair by $\dfrac{R_1 R_2}{R_1 + R_2}$ (the "product over sum" form, valid for two resistors only), redraw, and repeat until one resistor is left. Two rules decide **which resistor dissipates the most power** without any arithmetic: components carrying the **same current** dissipate more in the **larger** resistance ($P = I^2R$); components with the **same p.d.** dissipate more in the **smaller** resistance ($P = V^2/R$). In a series–parallel mix the single resistor that carries the **whole** current usually dissipates the most, because a parallel branch carries only a share of it.

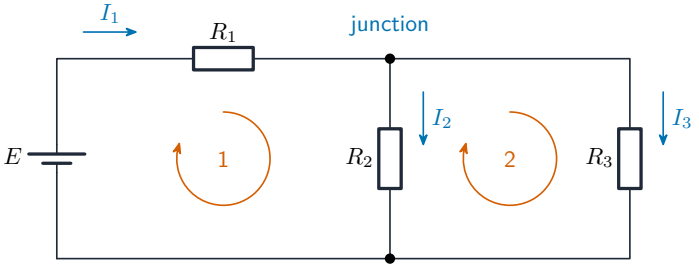
> **Worked example.** Three resistors, each of resistance R, are connected with two in series and that pair in parallel with the third. The total resistance between the ends is 8.0 Ω. Find R.

The series pair is $2R$; in parallel with R: $R_{\mathrm{T}} = \dfrac{2R \times R}{2R + R} = \dfrac{2R}{3}$. So $\dfrac{2R}{3} = 8.0$, giving $R = 12\ \Omega$. Check: the single resistor R carries twice the current of the pair (same p.d., half the resistance), so it dissipates the most power.

Solving a circuit

1. **Label** every current with a symbol and a chosen direction.
2. Use **Kirchhoff's first law** at each junction to link the currents.
3. Use **Kirchhoff's second law** around each loop to get equations in the p.d.s.
4. **Use** $V = IR$ for each resistor.
5. Solve the equations together.

For symmetric resistor networks, use the symmetry to spot branches with equal currents – the branch with the most current gives the most power ($P = I^2R$).



The standard two-loop circuit: one junction equation and two loop equations fix all three currents

Worked example. In the circuit above $\varepsilon = 12$ V (negligible internal resistance), $R_1 = 2.0\ \Omega$, $R_2 = 6.0\ \Omega$ and $R_3 = 3.0\ \Omega$. Find the three currents.

Kirchhoff's method. Junction: $I_1 = I_2 + I_3$. Loop 1 (through ε, R_1, R_2): $12 = 2.0I_1 + 6.0I_2$. Loop 2 (through R_2 and R_3, no source): $0 = 6.0I_2 - 3.0I_3$, so $I_3 = 2I_2$. Then $I_1 = 3I_2$ and $12 = 6.0I_2 + 6.0I_2$, giving $I_2 = 1.0$ A, $I_3 = 2.0$ A and $I_1 = 3.0$ A.

Reduction check. 6.0 Ω and 3.0 Ω in parallel give 2.0 Ω; with R_1 the total is 4.0 Ω, so $I_1 = 12/4.0 = 3.0$ A and the p.d. across the parallel pair is $3.0 \times 2.0 = 6.0$ V, giving $I_2 = 6.0/6.0 = 1.0$ A and $I_3 = 6.0/3.0 = 2.0$ A. Both methods must agree; a negative answer for a current simply means you guessed its direction the wrong way.

Potential dividers

A **potential divider** is two (or more) resistors in series across a source. The p.d. across each resistor is in **direct proportion** to its resistance:

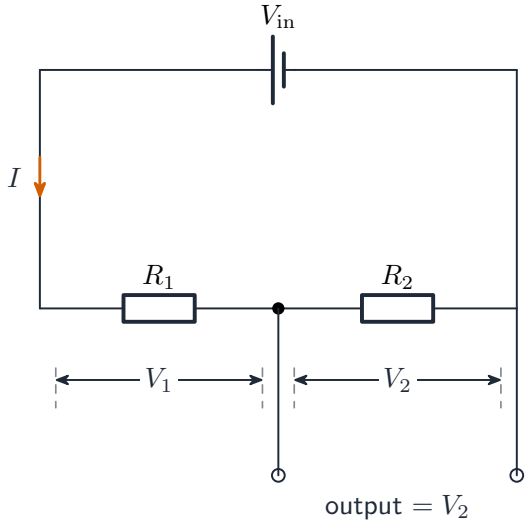
$$V_1 = V_{\text{in}} \cdot \frac{R_1}{R_1 + R_2}, \qquad V_2 = V_{\text{in}} \cdot \frac{R_2}{R_1 + R_2}.$$

The output (tapped between R_1 and R_2) can be set to any **voltage** between 0 and V_{in} by choosing the resistances. A **potentiometer** used with all three of its connections (a slider on a uniform-resistance track) gives a smoothly variable divider; the same component with only two connections is a **rheostat**, which just changes the current.

Worked example. A 6.0 V supply is connected across a 2.0 kΩ resistor in series with a 4.0 kΩ resistor. Find the output voltage tapped across the 4.0 kΩ resistor.

$$V_2 = V_{\text{in}} \cdot \frac{R_2}{R_1 + R_2} = 6.0 \times \frac{4.0}{2.0 + 4.0} = 4.0\ \text{V}.$$

The same answer comes from the **current-first** route the mark scheme often lays out: $I = V_{\text{in}}/(R_1 + R_2) = 6.0/6000 = 1.0$ mA, then $V_2 = IR_2 = 1.0 \times 10^{-3} \times 4000 = 4.0$ V. Two consequences worth remembering: the **larger** resistance takes the **larger** share of the p.d.; and the ratio formula holds only while **no current is drawn from the output** – a load (or a low-resistance voltmeter) in parallel with R_2 lowers the effective resistance of R_2 and so lowers V_2.



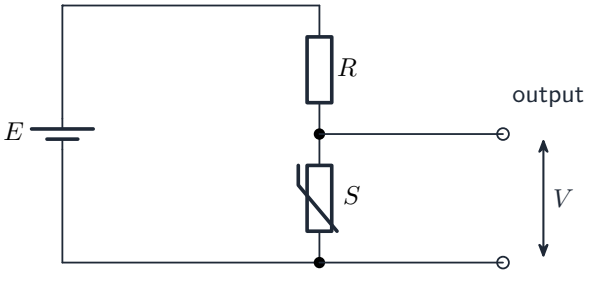
A potential divider – the p.d. splits between R1 and R2 in proportion to their resistances

Sensor circuits

Replace one fixed resistor with a **sensor** whose resistance changes with a physical quantity:

- **thermistor** (NTC): R falls as temperature rises. In a divider, the output voltage changes with temperature in a fixed direction.
- **light-dependent resistor** (LDR): R falls as **light intensity** rises, giving a brightness-dependent output.

Connect the output to a **transistor** base or a **comparator** to switch a load on or off when the temperature or light passes a **threshold** .

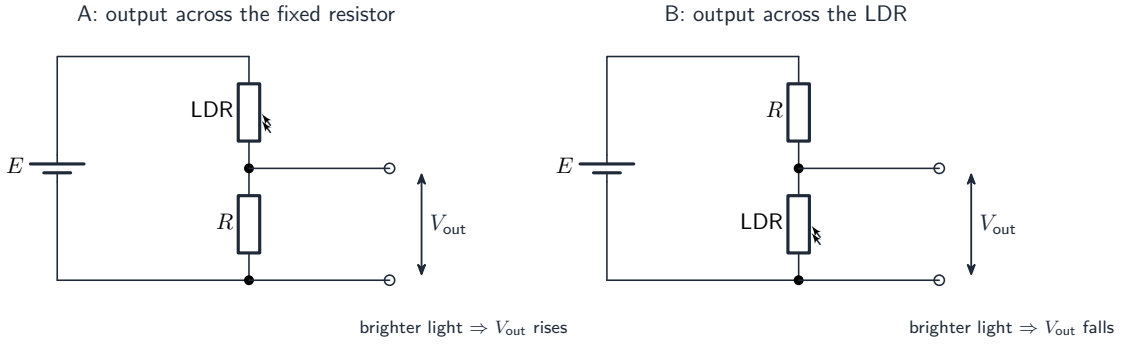


A thermistor in a potential divider gives an output voltage that changes with temperature

Which way does the output go? The output can be taken across the **sensor** or across the **fixed resistor**, and the two choices respond in opposite directions. Argue through the current, in this order, and you have the full three-mark answer:

1. the light gets brighter (or the temperature rises), so the resistance of the LDR (or thermistor) **falls**;
2. the total resistance of the series circuit falls, so the **current rises** ($I = \varepsilon/R_{\text{total}}$);
3. the p.d. across the **fixed resistor** $= IR$ **rises**, so the p.d. across the **sensor** $= \varepsilon - IR$ **falls**.

So an output across the fixed resistor **rises** with light or temperature; an output across the sensor **falls**. If the source has internal resistance, add one more link: the larger current also increases Ir, so the terminal p.d. falls slightly.



Where you take the output decides the direction of the change: across the fixed resistor it rises with light, across the LDR it falls

Worked example. A battery of e.m.f. 9.0 V and negligible internal resistance is connected in series with an LDR and a 1200 Ω resistor. In the light the LDR has a resistance of 1800 Ω. Calculate the p.d. across the LDR, and state and explain what happens to it when the light intensity decreases.

$V_{\text{LDR}} = 9.0 \times \dfrac{1800}{1800 + 1200} = 5.4$ V. Less light → the resistance of the LDR increases → the total resistance increases and the current **decreases** → the p.d. across the 1200 Ω resistor (IR) decreases → the p.d. across the LDR ($9.0 - IR$) **increases**.

Worked example. A thermistor in series with a 5800 Ω resistor across a 6.0 V supply gives a p.d. of 2.9 V across the resistor. Find the resistance of the thermistor.

Current $= 2.9/5800 = 5.0 \times 10^{-4}$ A. The thermistor takes the remaining $6.0 - 2.9 = 3.1$ V, so $R = 3.1/(5.0 \times 10^{-4}) = 6200$ Ω (or, by ratio, $R = 5800 \times 3.1/2.9$).

■ Potentiometer and the null method

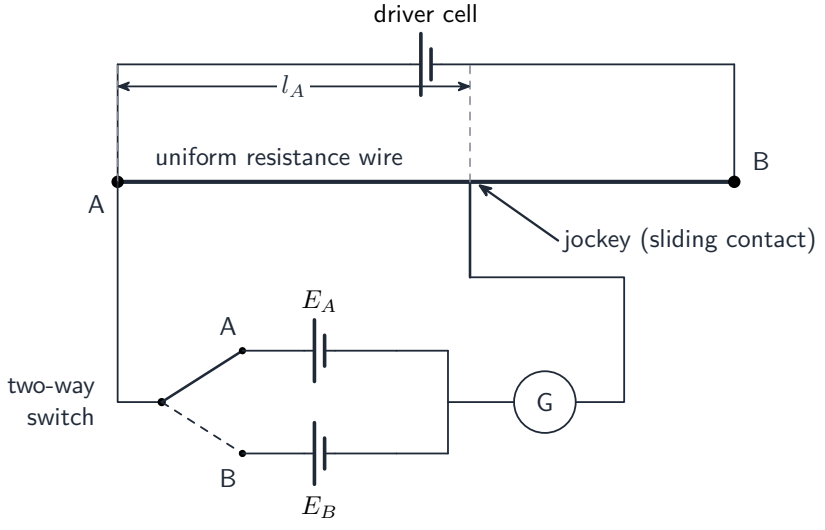
A **potentiometer** is a uniform resistance wire of length L_0 with a sliding contact (**jockey**). The resistance per unit length is uniform, so the p.d. from one end to the jockey is proportional to the length:

$$V_x = V_{\text{full}} \cdot \frac{x}{L_0}.$$

To **compare two e.m.f.s** (an unknown cell against a standard cell), connect each in turn with the jockey through a **galvanometer** . Slide the jockey until the galvanometer reads zero (a **null** – no current flows through the cell being measured, because the potentiometer's voltage there exactly opposes the cell's e.m.f.). The **balance length** – the wire length from the end to the jockey at balance – is proportional to the e.m.f. being measured, so the two lengths are in the ratio of the e.m.f.s:

$$\frac{\varepsilon_1}{\varepsilon_2} = \frac{l_1}{l_2}.$$

This is a **null method** : you find the balance (zero current) instead of measuring a current's value. Its advantage is that at balance the unknown cell gives no current, so its internal resistance does not affect the result.



A potentiometer comparing two cell e.m.f.s by the null method

Why the null method is better than a voltmeter. At balance no current is drawn from the cell being measured, so there is no Ir drop inside it: the balance point measures the **e.m.f.**, not the terminal p.d. A voltmeter always draws some current, so it reads slightly less than the e.m.f.

Reading the balance point. The p.d. per unit length of the wire is fixed by the **driver cell** and the resistance of the wire. Anything that makes the p.d. being balanced **larger** moves the balance point **further** along the wire; anything that makes the p.d. per unit length larger (a driver cell of larger e.m.f., or a wire that takes a larger share of the driver's p.d.) makes the balance length **shorter**. Replacing the wire by one of the same length but **greater diameter** lowers its resistance ($R = \rho L/A$): if the driver cell has negligible internal resistance and nothing else is in series, the p.d. across the wire is still the full e.m.f. and the balance length does not move; if the

driver has internal resistance or a series resistor, the wire's share of the e.m.f. falls, the p.d. per metre falls, and the balance length grows. Say which case you are in.

> **Worked example.** A potentiometer wire XY of length 2.0 m carries a p.d. of 1.50 V. A cell of e.m.f. E is balanced when the jockey is 1.6 m from X. Find E, and state what happens to the balance length if the driver cell is replaced by one of larger e.m.f.

$E = 1.50 \times \dfrac{1.6}{2.0} = 1.2$ V. A larger driver e.m.f. gives a larger p.d. per metre, so the same 1.2 V is reached at a **shorter** length: the jockey must move towards X.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
electromotive force (e.m.f.)	the energy transferred per unit charge by a source in driving charge round a complete circuit
potential difference (p.d.)	the energy transferred per unit charge from electrical energy to other forms of energy
internal resistance	the resistance to current inside a source of e.m.f., which causes a p.d. Ir across the source when a current flows
terminal p.d.	the p.d. across the terminals of a source, equal to e.m.f. $-Ir$
Kirchhoff's first law	the sum of the currents into a junction is equal to the sum of the currents out of the junction (conservation of charge)
Kirchhoff's second law	the sum of the e.m.f.s round a closed loop is equal to the sum of the p.d.s round the loop (conservation of energy)
potential divider	two or more resistors in series across a supply, giving a p.d. across one of them that is a fraction of the supply p.d.
potentiometer	a uniform resistance wire with a sliding contact, used to compare p.d.s by finding the length at which a galvanometer reads zero

Term	Definition
null method	a measurement made by adjusting a circuit until a meter reads zero, so no current is drawn from the component being measured
balance length	the length of potentiometer wire between one end and the sliding contact when the galvanometer reads zero

Exam tips

- Apply **Kirchhoff's laws**: current into a junction = current out (charge conserved); $\sum$ e.m.f. = $\sum$ p.d. round a loop (energy conserved). Name the law you are using when a question says "use Kirchhoff's laws".
- Combine resistors: **series** $R = R_1 + R_2$; parallel $1/R = 1/R_1 + 1/R_2$. A parallel combination is always smaller than its smallest resistor.
- A **potential divider** splits voltage in the ratio of the resistances; the larger resistance takes the larger p.d.
- Include **internal resistance**: e.m.f. $= I(R + r)$ – the "lost volts" are Ir. When a circuit changes, argue resistance $\rightarrow$ current in the cell $\rightarrow Ir \rightarrow$ terminal p.d.
- "Negligible internal resistance" means the terminal p.d. **is** the e.m.f., whatever the current. Look for the phrase before you start.
- Read a V–I graph of a source as $V = \varepsilon - Ir$: intercept ε, gradient $-r$, and the current at $V = 0$ is the maximum (short-circuit) current ε/r.
- In Paper 5, rearrange the circuit relation into $y = mx + c$ before plotting: a cell of e.m.f. ε and internal resistance r feeding n equal resistors R in parallel obeys $\varepsilon = I\left(\dfrac{R}{n} + r\right)$, so $\dfrac{1}{I} = \dfrac{R}{\varepsilon} \cdot \dfrac{1}{n} + \dfrac{r}{\varepsilon}$ – a graph of $1/I$ against $1/n$ has gradient R/ε and intercept r/ε.

Common mistakes

- Writing $V = IR$ with the e.m.f. and the external resistance only, forgetting r. Use $\varepsilon = I(R + r)$.
- Saying e.m.f. is "the force that pushes the charge". It is energy per unit charge; the volt is a joule per coulomb.
- Pairing the laws with the wrong conservation law. First law – charge; second law – energy.
- Using "product over sum" for three parallel resistors. It works for two only; otherwise add the reciprocals.

- Explaining a sensor circuit by ”the resistance changes so the voltage changes”. The marks are for the chain: resistance → total resistance → current → IR across the fixed resistor → the rest across the sensor.
- Claiming a potentiometer at balance “draws no current from the driver cell”. It draws none from the **cell being measured**; the driver cell always supplies the wire current.
- Rounding a ”show that” value before the last line – write 1.40 V, then say it is 1.4 V.

Particle physics

A-Level Physics

The nuclear atom

Geiger–Marsden α-particle scattering

Alpha particles α fired at a thin gold foil were seen to:

- mostly pass straight through, with very little **deflection**,
- sometimes deflect through small angles,
- rarely (about 1 in 8000) deflect through angles greater than 90°.

From this Rutherford worked out:

- the atom is **mostly empty space** (most α-particles pass straight through),
- there is a tiny, dense, positively charged **nucleus** at the centre (the rare large deflections need a concentrated charge to push the α away),
- almost all of the atom's mass is in this nucleus.

Order of magnitude: atom diameter $\sim 10^{-10}$ m, nucleus diameter $\sim 10^{-15}$ m – the nucleus is about 10^5 times smaller than the atom.

The three results and what each proves – the mark scheme pairs them exactly like this:

Observation	Conclusion
the vast majority pass straight through with little or no deflection	most of the atom is empty space
a small number are deflected through small angles	there is a concentrated positive charge (the nucleus) that repels the positive α-particles
a very small minority (about 1 in 8000) are deflected through more than 90°, some straight back	the nucleus is very small compared with the atom, and contains almost all of the atom's mass

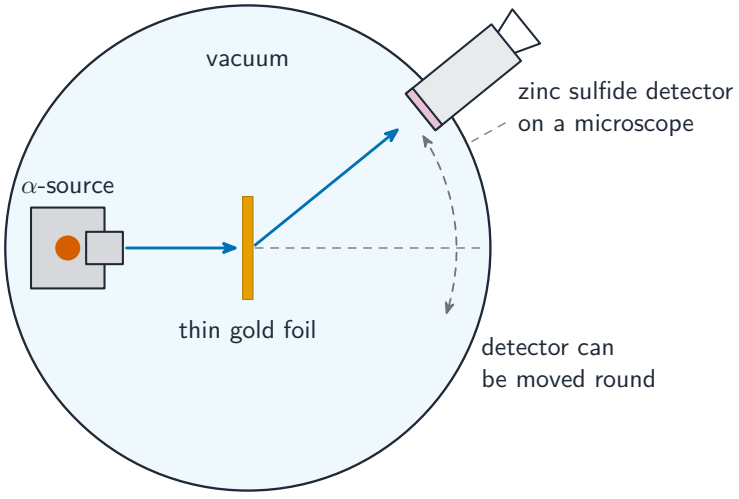
Three details of the experiment are also asked: the foil is **thin** so that each α-particle meets at most one nucleus (and is not absorbed); the chamber is a **vacuum** so the α-particles are not stopped or scattered by air; gold is used because it can be beaten into a very thin sheet and has a heavy, highly charged nucleus.

> **Worked example.** Explain why a very small minority of the α-particles are scattered through angles greater than 90°.

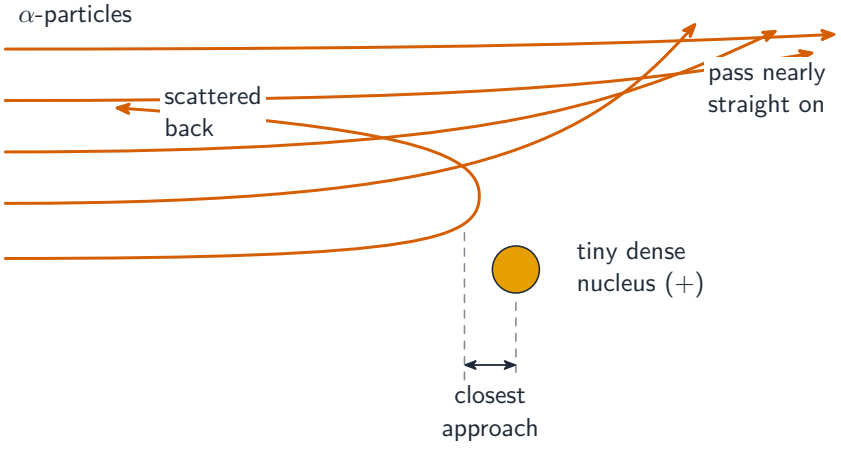
Only an α-particle that approaches a nucleus almost **head-on** is turned back. The nucleus and the α-particle are both positive, so the **electrostatic** repulsion is very large at small separation, and the nucleus is much more massive than the α-

particle, so it is the α-particle that is turned round. Because the nucleus is tiny, very few α-particles get that close – hence the small minority. As the α-particle approaches, its kinetic energy is converted to electric potential energy; at the point of closest approach it is momentarily at rest and all of its kinetic energy has become potential energy.

The deflection depends on how close the path passes to the nucleus: the smaller the distance, the larger the angle, and the paths are symmetrical about the line through the nucleus.



The α-scattering experiment – α-particles strike a thin gold foil in a vacuum



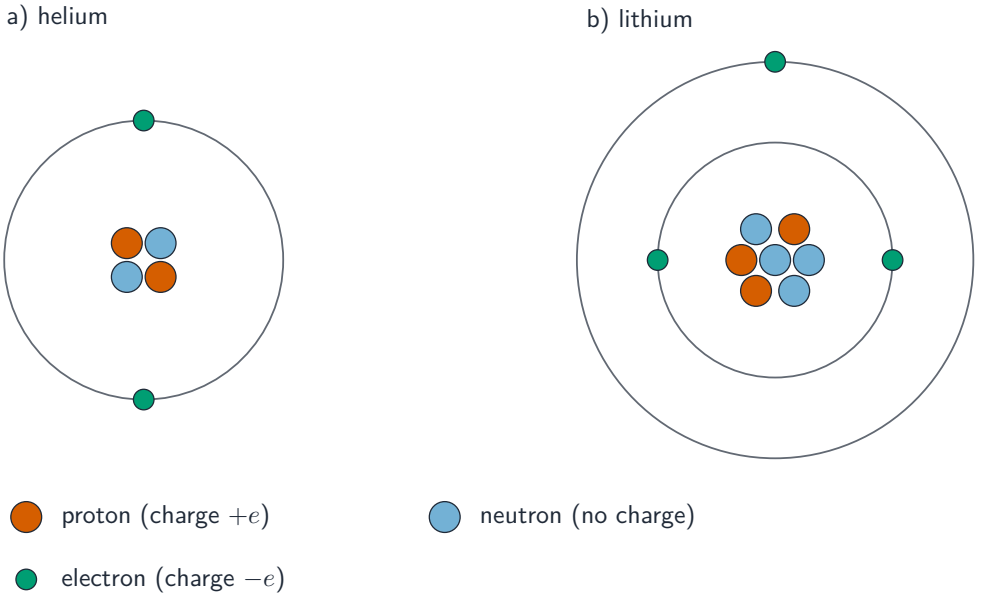
Most α-particles pass nearly straight through; a few are deflected sharply by the tiny nucleus

Simple nuclear model

An atom has:

- a central nucleus of **protons** (positive, charge $+e$) and **neutrons** (no charge),
- **electrons** (charge $-e$) around the nucleus.

The proton and neutron have almost the same mass ($\approx$ 1 u); the electron is about $\frac{1}{1836}$ of the proton's mass.



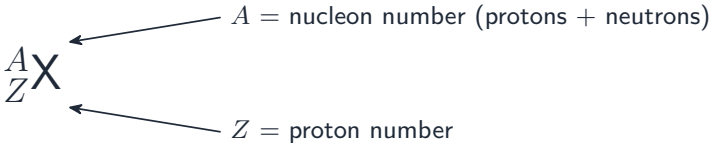
Simple models of a helium atom and a lithium atom (not to scale)

Worked example. Describe the structure of an atom of uranium-238, $^{238}_{92}\mathrm{U}$.

A nucleus containing 92 protons and $238 - 92 = 146$ neutrons, with 92 electrons in orbit around it. "Orbital electrons" is the syllabus phrase – the electrons are outside the nucleus, and the neutral atom has as many electrons as protons.

Notation and key numbers

For a **nuclide** written $^{A}_{Z}\mathrm{X}$:



Nuclide notation: nucleon number on top, proton number below

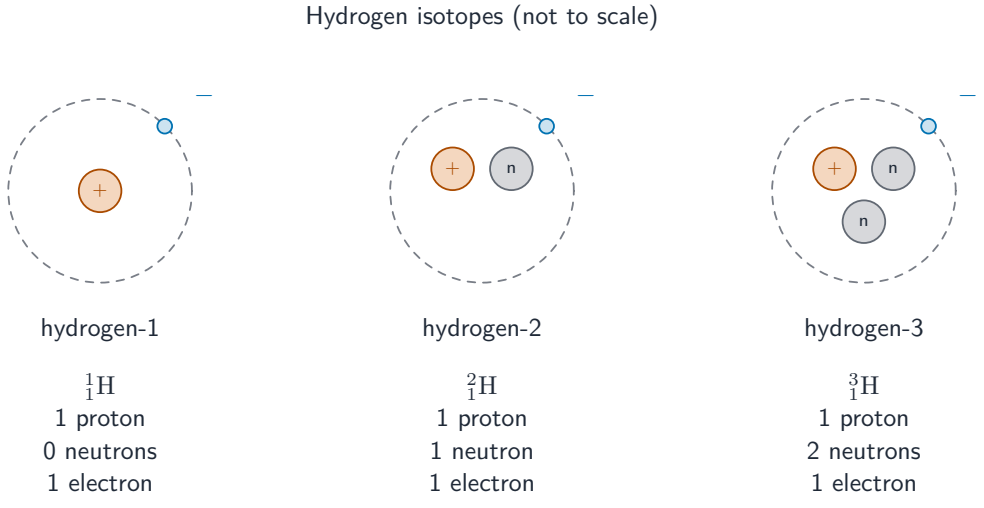
- **proton number** Z (also the atomic number): the number of protons. It fixes the element.
- **nucleon number** A (also the mass number): the total number of **nucleons** (protons + neutrons).
- number of neutrons $N = A - Z$.

A neutral atom has the same number of electrons as protons.

Isotopes

Isotopes are atoms of the same element (same Z) with different numbers of neutrons (different A). They behave the same chemically but differently in the nucleus. Example: ${}^{12}_{6}\mathrm{C}$ and ${}^{14}_{6}\mathrm{C}$ are isotopes of carbon.

The examiner's wording: isotopes are nuclei (or atoms) with the **same number of protons** but **different numbers of neutrons** – equivalently the same proton number Z and different nucleon numbers A. Hydrogen has three:



The three isotopes of hydrogen: the same one proton, different numbers of neutrons

> **Worked example.** Tritium, ${}^{3}_{1}\mathrm{H}$, is an isotope of hydrogen. State the numbers of protons, neutrons and electrons in a neutral tritium atom, and give the quark composition of its nucleus.

$Z = 1$ proton; $A - Z = 3 - 1 = 2$ neutrons; 1 electron (neutral, so electrons = protons). The nucleus is one proton (uud) and two neutrons (udd, udd): **4 up quarks and 5 down quarks**. Charge check: $4 \times \frac{2}{3} - 5 \times \frac{1}{3} = +1$, the proton number. A "labelled diagram" of the atom is the nucleus (1 p, 2 n) with one electron orbiting outside it.

Conservation laws in nuclear processes

In any nuclear process:

- nucleon number A is conserved (total A before = total A after),
- **charge** is conserved (this is **conservation of charge**).

These two rules let you balance decay and reaction equations.

Unified atomic mass unit

The **unified atomic mass unit** , symbol u, is set so that an atom of $^{12}_{6}\mathrm{C}$ has mass exactly 12 u. Numerically,

$$1\ \mathrm{u} = 1.661 \times 10^{-27}\ \mathrm{kg}.$$

A proton has mass ≈ 1.007 u; a neutron ≈ 1.009 u; an electron $\approx 5.5 \times 10^{-4}$ u.

> **Worked example.** A nucleus X has 14 nucleons and p protons. Its charge-to-mass ratio is 4.1×10^{7} C kg^{-1}. Find p and identify X.

Charge $= pe$; mass ≈ 14 u. So $\dfrac{pe}{14\mathrm{u}} = 4.1\times10^{7}$, giving $p = \dfrac{4.1 \times 10^{7} \times 14 \times 1.66 \times 10^{-27}}{1.60 \times 10^{-19}} = 5.96 \approx 6$. Six protons and 14 nucleons: X is carbon-14, $^{14}_{6}\mathrm{C}$. Keep every figure until the end, then round to the nearest whole number of protons.

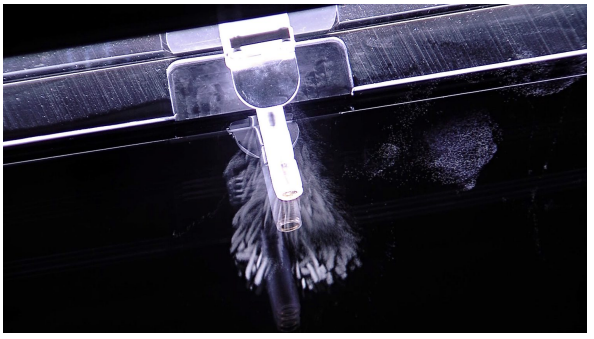
Radioactive emissions

An unstable nucleus rearranges itself and gives out one of three kinds of **radiation** . This is **radioactive** decay. Each kind has its own properties.

α-radiation

- **Made of**: a helium-4 nucleus, $^{4}_{2}\alpha$ (two protons + two neutrons).
- **Mass**: ≈ 4 u.
- **Charge**: $+2e$.
- **Range in air**: a few cm. Stopped by a sheet of paper.
- **Ionising power**: strong – it is good at **ionising** .
- **Energy spectrum**: **discrete** (one decay gives α-particles at one or a few sharp energies).

A **cloud chamber** makes the tracks visible: each α-particle leaves a short, straight, thick trail of tiny droplets as it ionises the air. The short equal lengths show the α-particles all carry about the same energy.



Alpha-particle tracks in a cloud chamber, fanning out from an americium-241 source

■ β-radiation

Two types of **beta particle** β :

- β^-: a fast electron, given out when a neutron turns into a proton.
- β^+: a **positron** (the electron's antiparticle), given out when a proton in a proton-rich nucleus turns into a neutron.

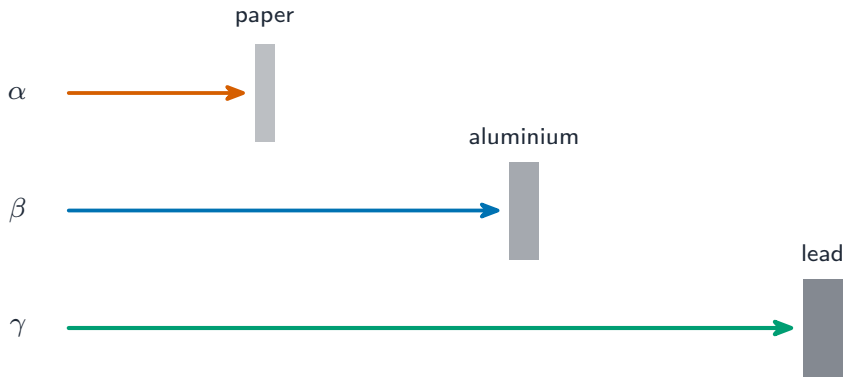
Properties (both types):

- **Mass**: $\approx 1/1836$ u (much less than α).
- **Charge**: $-e$ for β^-, $+e$ for β^+.
- **Range in air**: about 1 m. Stopped by a few mm of aluminium.
- **Energy spectrum**: **continuous** up to a maximum (see below).

■ γ-radiation

- **Made of**: a high-energy **photon** – part of the **electromagnetic spectrum** .
- **Mass**: zero (rest mass).
- **Charge**: zero.
- **Range in air**: large (follows the inverse-square law). Strongly **attenuated** by several cm of lead or about a metre of concrete.
- **Ionising power**: weakest.
- **Energy spectrum**: discrete (a **gamma ray** γ is given out as the nucleus drops between two nuclear energy levels).

A nucleus often gives out a γ-photon as a "tidy-up" step after an α or β decay leaves the **daughter nucleus** in an **excited state** .



Penetrating power: α is stopped by paper, β by aluminium, γ only attenuated by lead

Mass and charge, side by side – a table question asks for exactly these, in units of u and e:

Particle	Mass	Charge
proton	1 u (1.007 u)	$+e$
neutron	1 u (1.009 u)	0
electron, β^-	$\frac{1}{1836}$ u (5.5×10^{-4} u)	$-e$
positron, β^+	$\frac{1}{1836}$ u	$+e$
α-particle	4 u	$+2e$
γ-photon	0	0
neutrino, antineutrino	almost zero	0

> **Worked example.** Compare an α-particle with a β^+ particle in terms of their masses and charges (3 marks).

Both are positively charged, but the α-particle's charge ($+2e$) is twice that of the β^+ ($+e$). The α-particle's mass (4 u) is about 7300 times the mass of the β^+ (5.5×10^{-4} u). Give a ratio, not just "heavier": *twice the charge* and *about 7000 times the mass* are the marking points.

Because they carry charge, α- and β-particles are deflected by electric and magnetic fields – in opposite directions for opposite signs, the light β far more than the heavy α – while γ-rays, being uncharged, pass straight through.

Antiparticles, neutrinos and antineutrinos

Every particle has an **antiparticle** with the **same mass** but **opposite charge**. The positron is the antiparticle of the electron.

In β-decay, a third particle is always given out as well:

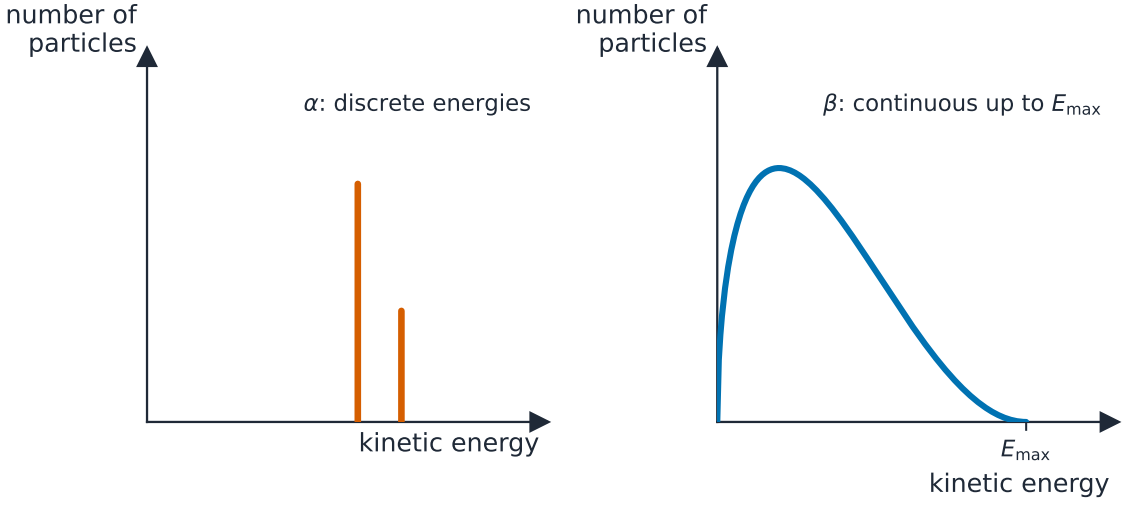
- β^- decay: an **antineutrino** $\bar{\nu}_e$.
- β^+ decay: a **neutrino** ν_e.

Neutrinos and antineutrinos have zero charge, very small mass, and barely interact – they are very hard to detect, but they must be there to balance energy, **momentum** and other conserved quantities in β-decay.

Why β has a continuous spectrum (and α does not)

In α-decay the **energy** released is shared between just two particles (the daughter nucleus and the α). Conservation of momentum and energy then fixes the α's energy to one value (discrete).

In β-decay the energy is shared between **three** particles (the daughter nucleus, the β, and the (anti)neutrino). The β can take any share from zero up to a maximum, so its energy spectrum is continuous.



α-particles from one decay have fixed (discrete) energies; β-particles share the energy with an (anti)neutrino, so their spectrum is continuous up to a maximum

> **Worked example.** The energy spectrum of the β^- particles from a source is continuous, from zero up to a maximum. Explain why (3 marks).

Each decay releases a fixed amount of energy. That energy is shared between the β^- particle, the **antineutrino** and the recoiling daughter nucleus. The antineutrino can take any share, so the β^- particle is left with any energy from zero up to the maximum – the maximum being when the antineutrino carries away almost none. (For an α-particle there are only two bodies, so momentum conservation fixes the share and the energy is discrete.)

■ Writing decay equations

A general α-decay:

$$^{A}_{Z}\mathrm{X} \rightarrow {}^{A-4}_{Z-2}\mathrm{Y} + {}^{4}_{2}\alpha$$

(check: $A = (A-4)+4$, $Z = (Z-2)+2$.)

A general β^- decay:

$$^{A}_{Z}\mathrm{X} \rightarrow {}^{A}_{Z+1}\mathrm{Y} + {}^{0}_{-1}\beta + \bar{\nu}_\mathrm{e}.$$

A general β^+ decay:

$$^{A}_{Z}\mathrm{X} \rightarrow {}^{A}_{Z-1}\mathrm{Y} + {}^{0}_{+1}\beta + \nu_\mathrm{e}.$$

> **Worked example.** Uranium-238, $^{238}_{92}\mathrm{U}$, decays by α-emission; carbon-14, $^{14}_{6}\mathrm{C}$, decays by β^--emission. Find each daughter nuclide.

α-decay lowers A by 4 and Z by 2; β^--decay leaves A unchanged and raises Z by 1:

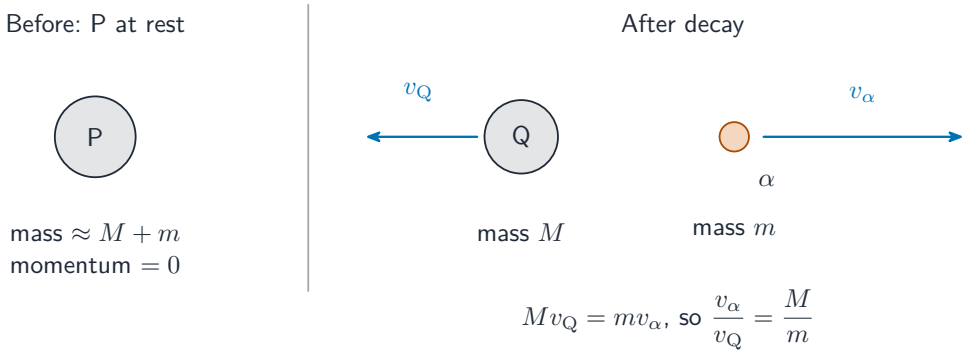
$$^{238}_{92}\mathrm{U} \rightarrow {}^{234}_{90}\mathrm{Th} + {}^{4}_{2}\alpha, \qquad {}^{14}_{6}\mathrm{C} \rightarrow {}^{14}_{7}\mathrm{N} + {}^{0}_{-1}\beta + \bar{\nu}_\mathrm{e}.$$

Determine quantitatively the changes. α-emission: A decreases by 4, Z decreases by 2. β^--emission: A unchanged, Z increases by 1. β^+-emission: A unchanged, Z decreases by 1. γ-emission: no change in either. A decay chain that ends at a known nuclide is solved from these: the number of α-decays is $(A_\mathrm{start} - A_\mathrm{end})/4$, and then the number of β^--decays makes Z come out right.

> **Worked example.** Thorium-230, $^{230}_{90}\mathrm{Th}$, decays in stages, by α- and β^--emission, to lead-206, $^{206}_{82}\mathrm{Pb}$. Find the number of each kind of decay.

α-decays: $(230-206)/4 = 6$. Six α-decays alone would lower Z by 12, to 78; the final Z is 82, so there are $82-78 = 4$ β^--decays.

A nucleus at rest recoils. When a stationary nucleus emits an α-particle, momentum is conserved: the daughter nucleus and the α-particle move off in **opposite directions** with momenta of **equal magnitude**. Since $p = mv$, the lighter α-particle moves much faster, and since $E_\mathrm{k} = p^2/2m$ it also takes most of the kinetic energy – the shares are in inverse proportion to the masses.

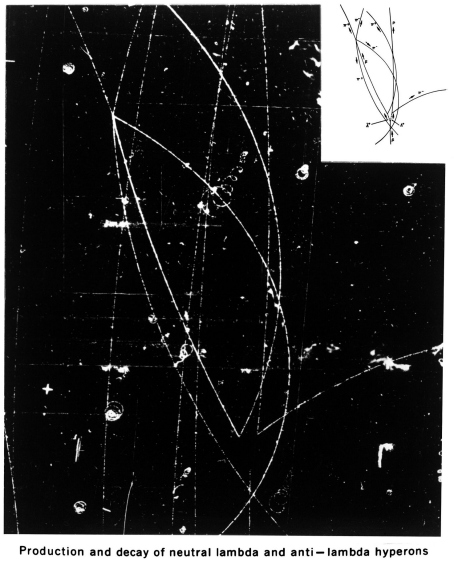


A nucleus at rest decays: the daughter recoils with momentum equal and opposite to the α-particle's

> **Worked example.** A stationary nucleus P of mass 243 u emits an α-particle of mass 4 u at 1.5×10^7 m s^{-1}. Find the speed of the daughter nucleus Q and the ratio of the kinetic energy of the α-particle to that of Q.

Q has mass $243-4 = 239$ u. Momentum: $239 v_\text{Q} = 4 \times 1.5 \times 10^7$, so $v_\text{Q} = 2.5 \times 10^5$ m s^{-1}, in the opposite direction to the α-particle. Kinetic energies: $\dfrac{E_\alpha}{E_\text{Q}} = \dfrac{p^2/2m_\alpha}{p^2/2m_\text{Q}} = \dfrac{m_\text{Q}}{m_\alpha} = \dfrac{239}{4} \approx 60$. The α-particle takes about 98% of the energy released.

Fundamental particles



A bubble chamber reveals the curved tracks of charged particles.



Where the particles come from: a linear accelerator uses electric fields to drive charged particles fast enough to probe the nucleus

Some particles are **fundamental particles** (point-like, with no smaller parts as far as we know); others are built from fundamental ones.

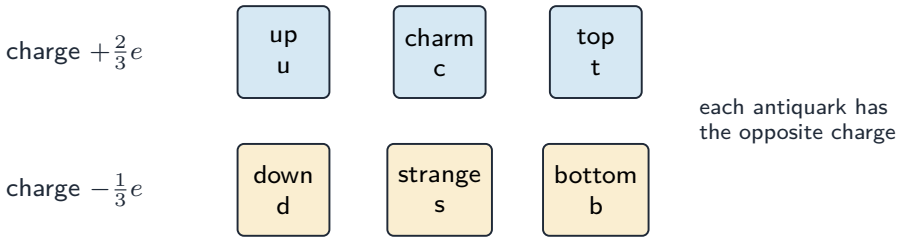
The one-mark definition: *a fundamental particle is one that cannot be broken down into smaller particles* (it has no internal structure). Electrons, neutrinos and quarks are fundamental; protons and neutrons are not.

Quarks

A **quark** is a fundamental particle. There are six **flavours** :

- **up** (u), charge $+\frac{2}{3}e$,
- **down** (d), charge $-\frac{1}{3}e$,
- **charm** (c), charge $+\frac{2}{3}e$,
- **strange** (s), charge $-\frac{1}{3}e$,
- **top** (t), charge $+\frac{2}{3}e$,
- **bottom** (b), charge $-\frac{1}{3}e$.

Each quark has an **antiquark** with the same size of charge but the opposite sign: $\bar{u}$ (charge $-\frac{2}{3}e$), $\bar{d}$ (charge $+\frac{1}{3}e$). No other quark property is tested.



The six quarks: up/charm/top carry $+\frac{2}{3}e$, down/strange/bottom carry $-\frac{1}{3}e$

Written as the table a question asks you to complete:

Quark	Charge	Antiquark	Charge
up (u), charm (c), top (t)	$+\frac{2}{3}e$	$\bar{\text{u}}$, $\bar{\text{c}}$, $\bar{\text{t}}$	$-\frac{2}{3}e$
down (d), strange (s), bottom (b)	$-\frac{1}{3}e$	$\bar{\text{d}}$, $\bar{\text{s}}$, $\bar{\text{b}}$	$+\frac{1}{3}e$

> **Worked example.** By reference to quark composition, show that the charge of a proton is $+1.6 \times 10^{-19}$ C.

A proton is uud: charge $= \frac{2}{3}e + \frac{2}{3}e - \frac{1}{3}e = +e = +1.6 \times 10^{-19}$ C. Write the three fractions and their sum – the mark is for the arithmetic, not the answer.

■ Hadrons: baryons and mesons

Particles built from quarks are **hadrons** . Two types:

- **baryons** – three quarks. Examples: proton (u u d), neutron (u d d). Charge check: $\frac{2}{3} + \frac{2}{3} - \frac{1}{3} = +1$ for the proton; $\frac{2}{3} - \frac{1}{3} - \frac{1}{3} = 0$ for the neutron.
- **mesons** – one quark and one antiquark (for example π^+ is $\text{u}\bar{\text{d}}$).

Protons and neutrons are **not** fundamental – they are baryons made of quarks.

"Compare baryons and mesons in terms of their constituent particles" (2 marks): both are hadrons made of quarks; a baryon is three quarks (or three antiquarks, for an antibaryon), a meson is one quark and one antiquark. Any charge you are given must come out of the quark charges:

> **Worked example.** (a) A meson has charge $-1e$. Give a possible quark composition. (b) A meson Q has charge 0. Give a possible composition. (c) A baryon is made of three quarks of different flavours, u, d and s. Find its charge. (d) Give the quark composition of an antineutron and its charge.

(a) $\text{d}\bar{\text{u}}$: $-\frac{1}{3} - \frac{2}{3} = -1$ ($\text{s}\bar{\text{u}}$ also works). (b) $\text{u}\bar{\text{u}}$: $+\frac{2}{3} - \frac{2}{3} = 0$ (or $\text{d}\bar{\text{d}}$). (c) uds: $\frac{2}{3} - \frac{1}{3} - \frac{1}{3} = 0$. (d) The antineutron is the antiparticle of udd, so it is $\bar{\text{u}}\bar{\text{d}}\bar{\text{d}}$: $-\frac{2}{3} + \frac{1}{3} + \frac{1}{3} = 0$ – the same mass and charge as the neutron, since both have charge zero.

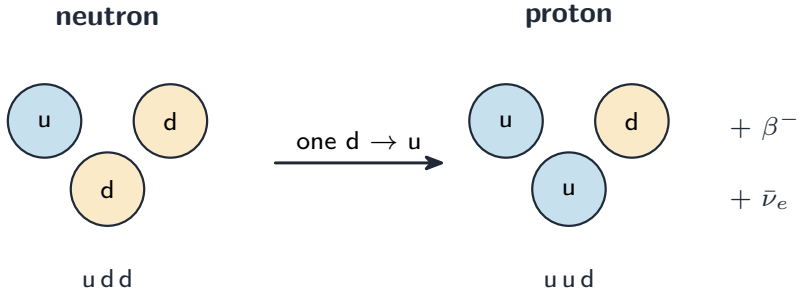
Quark changes in β-decay

In β^- decay a neutron turns into a proton; in quark terms, **one down quark turns into an up quark**:

$$\mathrm{d} \rightarrow \mathrm{u} + \beta^- + \bar{\nu}_\mathrm{e}.$$

In β^+ decay a proton turns into a neutron; **one up quark turns into a down quark**:

$$\mathrm{u} \rightarrow \mathrm{d} + \beta^+ + \nu_\mathrm{e}.$$



Beta-minus decay: one down quark becomes an up quark, turning a neutron into a proton

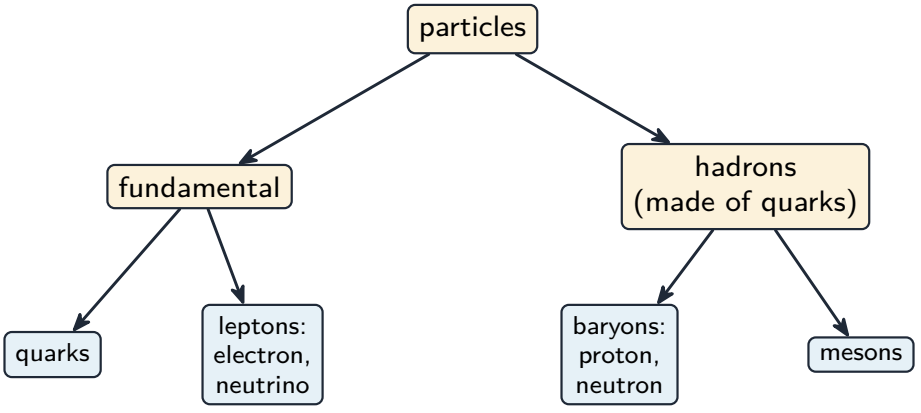
"Describe β^+ decay in terms of the fundamental particles involved" (2 marks): an up quark in a proton changes into a down quark, so the proton becomes a neutron, and a positron and an electron neutrino are emitted – $\mathrm{u} \rightarrow \mathrm{d} + \beta^+ + \nu_\mathrm{e}$. For β^- decay swap the roles: a down quark becomes an up quark, emitting an electron and an electron antineutrino. Name the (anti)neutrino: it is the "other lepton" the question asks for, and it is what makes the β energy continuous.

Leptons

Leptons are fundamental particles that are not made of quarks. The leptons you need are the **electron** and the **electron neutrino**, with their antiparticles the **positron** and the **electron antineutrino**. (Heavier leptons – the muon and tau – exist but are not tested.) Asked for "two different leptons", give *electron* and *neutrino*.

Classifying particles

To answer "which are fundamental?": **quarks** and **leptons** (electrons, positrons, neutrinos, antineutrinos) are fundamental; protons, neutrons, baryons, mesons and hadrons are **not** – they are built from quarks.



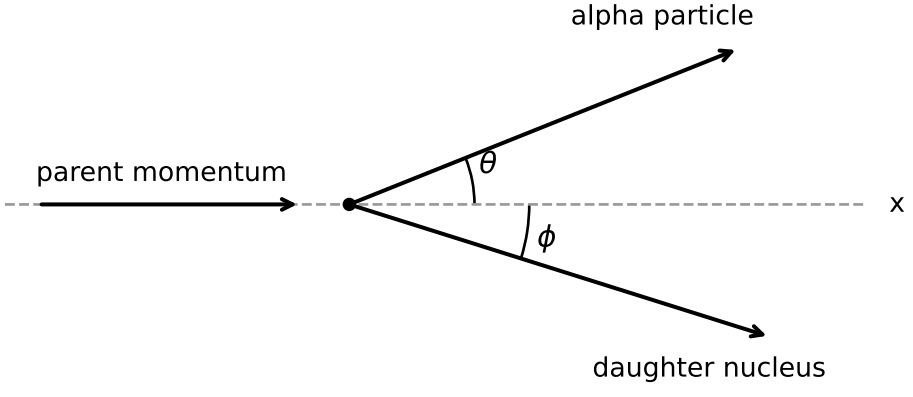
Fundamental particles (quarks, leptons) versus hadrons (baryons, mesons)

> **Worked example.** In the list – antineutrino, β^+ particle, neutron, positron, proton – underline the hadrons.

Neutron and proton (three quarks each). The antineutrino and the positron are leptons; the β^+ particle *is* a positron. The same list asked as "which are not fundamental?" has the same answer: neutron and proton.

Antimatter. Every particle has an antiparticle of the same mass and opposite charge; a positron and an electron are alike in mass (and in the size of their charge) and differ in the sign of their charge. An **antihydrogen** atom is an antiproton (charge $-e$, made of $\bar{u}\bar{u}\bar{d}$) with a positron in orbit.

A moving nucleus decays



Use conservation of momentum **along each axis separately**. Choose x along the parent's initial velocity, so the initial transverse momentum is zero. If the products move on opposite sides of x, their transverse momenta cancel: $m_\alpha v_\alpha \sin\theta = m_d v_d \sin\phi$. Along x, their momenta add to the parent's initial momentum.

Worked example. Products have momenta 2.0×10^{-20} and 1.0×10^{-20} kg m/s. The daughter moves 30° below x. Transverse conservation gives $\sin\theta = p_d \sin\phi / p_\alpha = (1.0\times10^{-20}\sin 30°)/(2.0\times10^{-20}) = 0.25$, so the alpha particle moves at 14.5° above x. The initial total momentum is not zero; only its transverse component is zero.

■ count quarks in a whole nucleus

A nucleus ${}^{A}_{Z}X$ has Z protons (uud) and $A - Z$ neutrons (udd). Thus $N_u = 2Z + (A - Z) = A + Z$ and $N_d = Z + 2(A - Z) = 2A - Z$. Check $N_u + N_d = 3A$ and $q = (2N_u - N_d)e/3 = Ze$.

Worked example. Carbon-14 has six protons and eight neutrons: $N_u = A + Z = 14 + 6 = 20$ and $N_d = 2A - Z = 28 - 6 = 22$. In beta-minus decay one down quark becomes up, so the daughter has 21 up and 21 down quarks.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
proton number Z	the number of protons in a nucleus
nucleon number A	the total number of protons and neutrons in a nucleus
isotopes	nuclei of the same element (same number of protons) with different numbers of neutrons
nuclide	a nucleus with a particular number of protons and a particular number of neutrons
unified atomic mass unit	one twelfth of the mass of an atom of carbon-12
antiparticle	a particle with the same mass as the corresponding particle but the opposite charge
fundamental particle	a particle that cannot be broken down into smaller particles
hadron	a particle made of quarks: a baryon (three quarks) or a meson (a quark and an antiquark)
lepton	a fundamental particle that is not made of quarks, such as the electron or the neutrino

Term	Definition
α-particle	a helium-4 nucleus: two protons and two neutrons, charge $+2e$, mass 4 u
β^- / β^+ particle	an electron / a positron emitted from a nucleus when a neutron becomes a proton / a proton becomes a neutron

Exam tips

- Describe the **nuclear atom** (a small, dense, positive nucleus) using the alpha-scattering evidence – pair each observation with its conclusion.
- Compare α, β and γ by **charge, mass, ionising power and penetration**; when asked to compare two particles, give the ratio ("twice the charge", "about 7000 times the mass").
- Use quark composition (proton *uud*, neutron *udd*) and check that **charge and nucleon number balance** in every equation you write.
- In β^- decay a neutron becomes a proton, an electron and an antineutrino; in β^+ decay a proton becomes a neutron, a positron and a neutrino. Name the (anti)neutrino every time.
- A nucleus at rest that decays gives its two products **equal and opposite momentum**, so the lighter one is faster and carries most of the kinetic energy.
- For a decay chain, count the α-decays from the change in A first, then fix Z with β-decays.

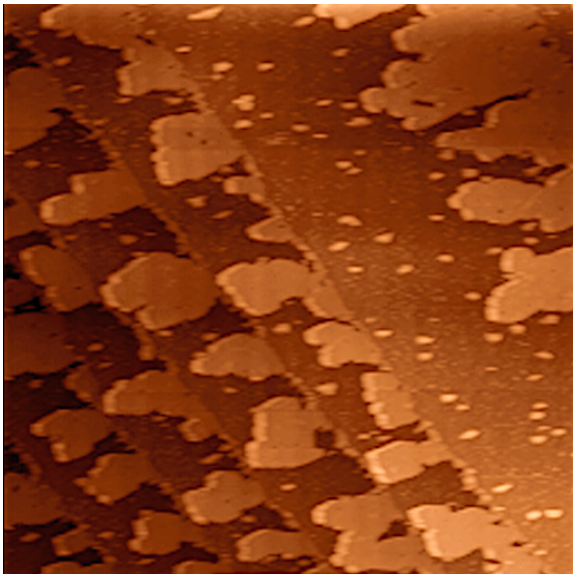
Common mistakes

- Concluding "the nucleus is positive" from the *straight-through* result. That result shows empty space; the large-angle deflections show the concentrated positive charge.
- Defining isotopes by "different mass" only. The mark needs *same number of protons, different number of neutrons.*
- Writing ${}^{0}_{-1}\beta$ for β^+ decay. A positron is ${}^{0}_{+1}\beta$, and Z goes *down* by one.
- Forgetting the (anti)neutrino in a β-decay equation, or putting a neutrino with β^-. β^- goes with the antineutrino, β^+ with the neutrino.
- Saying protons or neutrons are fundamental. Only quarks and leptons are.
- Giving a meson as "two quarks". It is a quark **and an antiquark**.
- Treating an antiquark's charge as the same sign as the quark's. $\bar{\text{u}}$ is $-\frac{2}{3}e$, $\bar{\text{d}}$ is $+\frac{1}{3}e$.

Atomic structure

A-Level Chemistry

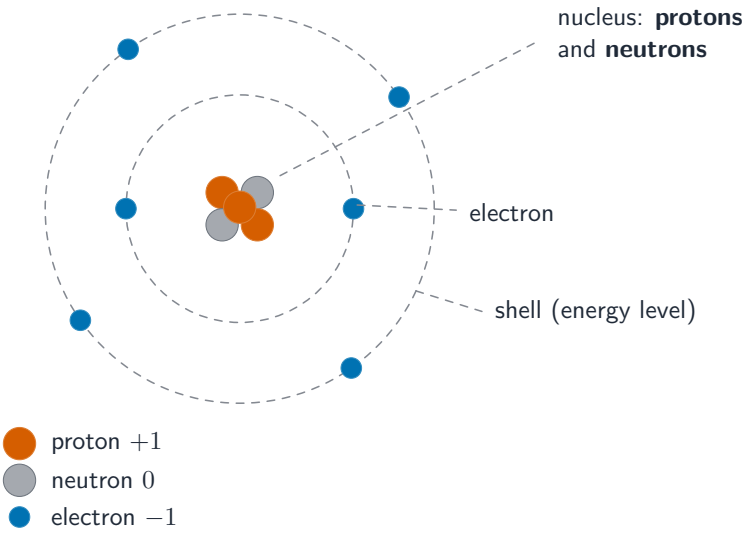
What an atom is made of



A scanning tunnelling microscope can image individual atoms.

Everything is made of **atoms** . An atom is mostly empty space. At its centre is a tiny, heavy **nucleus** . The nucleus holds two kinds of particle: **protons** and **neutrons** . Around the nucleus, in the empty space, move the **electrons** . The electrons stay in **shells** – layers at set distances from the nucleus.

The nucleus is very small but holds almost all the mass. The electrons take up almost all the space but have almost no mass.



An atom is mostly empty space: protons and neutrons sit in the tiny central nucleus, while electrons move in shells around it

Relative charge and relative mass

We compare the three particles using **relative charge** and **relative mass** . These are simple numbers, not real units.

Particle	Relative charge	Relative mass
proton	+1	1
neutron	0	1
electron	−1	$\frac{1}{1836}$ (about 0)

A proton and a neutron have almost the same mass. An electron is about 1836 times lighter. The proton is positive, the electron is negative, and the neutron has no charge – it is **neutral** .

Proton number and nucleon number

Two numbers describe the nucleus:

- the **proton number** (also called the **atomic number**), symbol Z – the number of protons.
- the **nucleon number** (also called the **mass number**), symbol A – the total number of protons and neutrons. Protons and neutrons are both **nucleons** .

So the number of neutrons is $A - Z$.

Counting particles in an atom or ion

For a neutral atom, the number of electrons equals the number of protons, which equals Z.

An **ion** is an atom that has lost or gained electrons, so it has a charge:

- a positive ion has fewer electrons than protons.
- a negative ion has more electrons than protons.

Example: $^{27}_{13}Al^{3+}$ has 13 protons, $27 - 13 = 14$ neutrons, and $13 - 3 = 10$ electrons (it lost 3 electrons to become 3+).

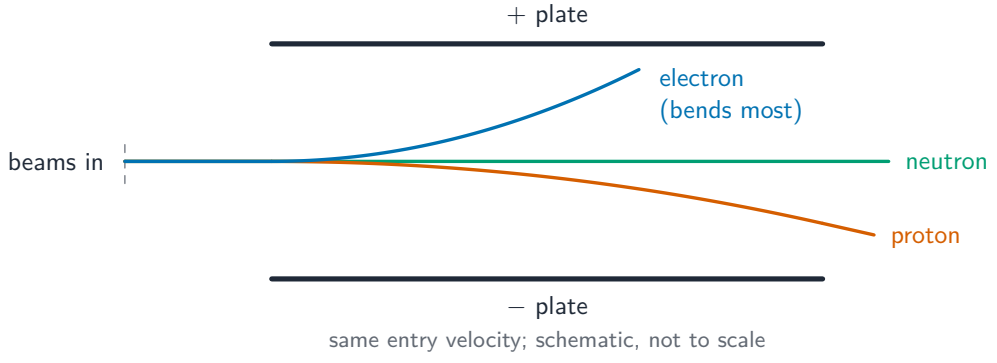
How mass and charge are spread out

Almost all the mass sits in the nucleus, because protons and neutrons are heavy and electrons are very light. All the positive charge is in the nucleus (the protons). The negative charge is spread out in the shells (the electrons).

Beams of particles in an electric field

Imagine beams of protons, neutrons and electrons moving at the same speed into an **electric field** between two charged plates:

- the proton beam bends towards the negative plate (protons are positive).
- the electron beam bends the other way, towards the positive plate. It bends much more, because the electron is far lighter – the same force gives a bigger **deflection** to a smaller mass.
- the neutron beam goes straight through. It has no charge, so the field gives it no force.



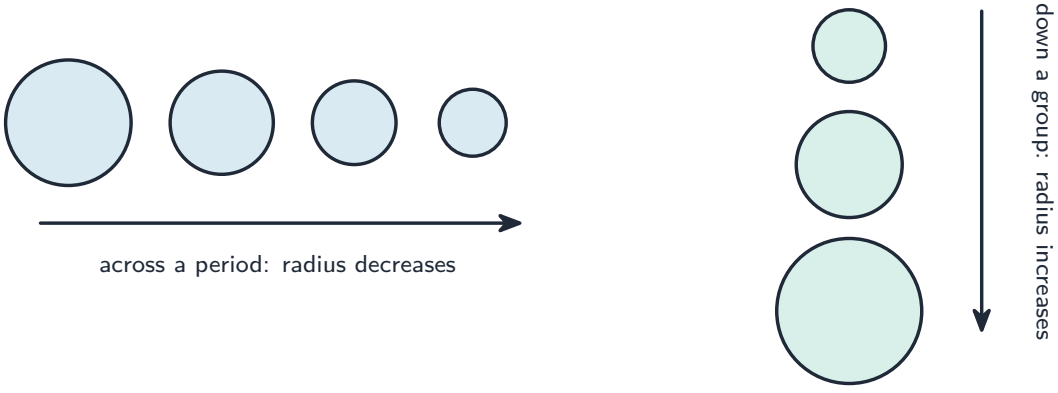
In an electric field the proton bends towards the − plate and the electron bends the opposite way and far more (it is much lighter); the neutron passes straight through

Atomic radius and ionic radius

The **atomic radius** is the size of an atom. The **ionic radius** is the size of an ion.

Across a **period** (left to right), the atomic radius gets **smaller**. The **nuclear charge** (the pull from the protons) rises, but the electrons go into the same outer shell, so the **shielding** by inner shells stays about the same. The stronger pull draws the outer shell inwards.

Down a **group** (top to bottom), the atomic radius gets **larger**. Each step down adds a new shell, so the outer electrons are further out and feel more shielding from the nucleus.



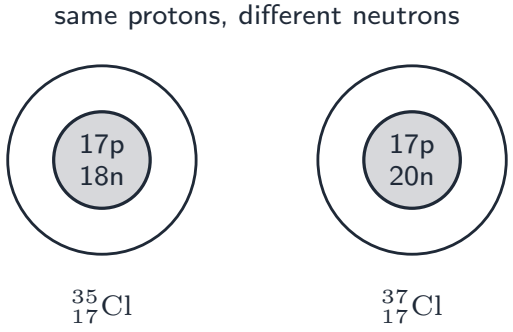
Across a period the atoms shrink (stronger nuclear pull on the same outer shell); down a group they grow (each step adds a shell)

For ions:

- a positive ion (**cation**) is smaller than its atom. Electron loss reduces electron-electron repulsion and increases the attraction for the remaining electrons. The whole outer occupied shell may be lost, but not every electron loss empties a shell.
- a negative ion (**anion**) is larger than its atom. It has gained electrons, so there is more **repulsion** between the electrons.
- among ions that have the same number of electrons, the one with more protons is smaller.

Isotopes

Isotopes are atoms of the same element with the same number of protons but a different number of neutrons. So isotopes have the same proton number Z but a different nucleon number A.



Two chlorine isotopes: same protons, different neutrons

We write an isotope as ${}^{A}_{Z}\mathrm{X}$: the nucleon number A on top, the proton number Z below. For example, chlorine has two main isotopes, ${}^{35}_{17}\mathrm{Cl}$ and ${}^{37}_{17}\mathrm{Cl}$.

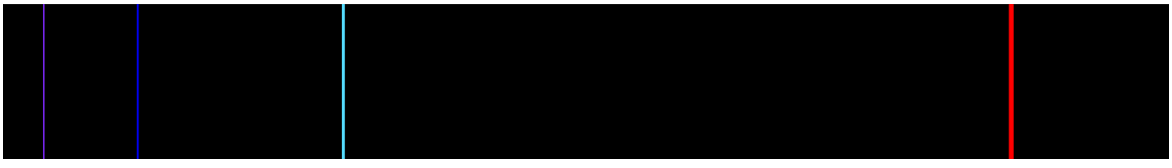
■ Same chemical properties

Chemical properties depend on the electrons, especially the outer electrons. Isotopes of one element have the same number of electrons arranged in the same way. So they react in exactly the same way – they have the same chemical properties.

■ Different physical properties

Some **physical properties** depend on mass, so they differ between isotopes. A heavier isotope has more neutrons, so more mass, and therefore a higher **density** . (The syllabus limits this difference to mass and density.)

Electrons, energy levels and orbitals



Hydrogen emits light only at discrete, characteristic wavelengths – its line emission spectrum.

Electrons are arranged in shells, sub-shells and orbitals.

■ Shells and the principal quantum number

Each shell is labelled by the **principal quantum number** $n = 1, 2, 3, \ldots$ A larger n generally means greater spatial extent, but n alone does not put all sub-shells of a multi-electron atom into energy order. The configurations and ionisation processes assessed in these sections are ground-state species from hydrogen to krypton.

Sub-shells and orbitals

Each shell is split into **sub-shells** , named s, p and d. Each sub-shell is built from **orbitals** . An orbital is a small region that can hold up to two electrons.

Sub-shell	Number of orbitals	Maximum electrons
s	1	2
p	3	6
d	5	10

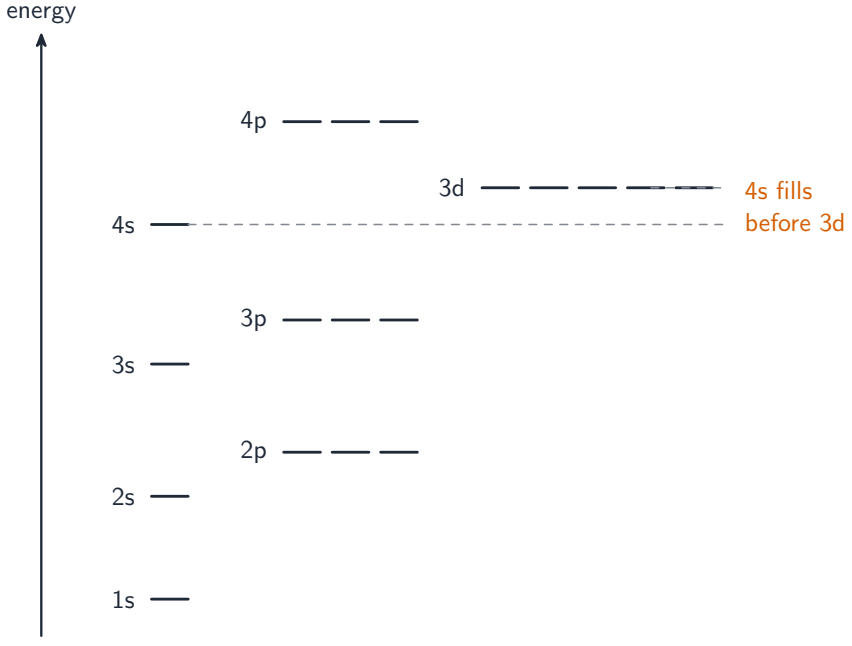
So an s sub-shell holds 2 electrons, a p sub-shell holds 6, and a d sub-shell holds 10.

Order of increasing energy

Electrons occupy the lowest-energy available arrangement. The usual neutral-atom filling sequence for the sub-shells studied here is:

$$1s < 2s < 2p < 3s < 3p < 4s < 3d < 4p$$

4s is reached before 3d in this usual construction sequence. The ordering is not fixed in every occupied atom or ion; apply the separate 4s-first removal rule for transition-metal cations.



The usual neutral filling order; it must not be used to claim that 4s is always lower than occupied 3d levels

Electronic configuration

The **electronic configuration** lists how many electrons are in each sub-shell. The lowest-energy arrangement is the **ground state** .

For iron (Fe, $Z = 26$):

$$1s^2\,2s^2\,2p^6\,3s^2\,3p^6\,3d^6\,4s^2$$

You can write a shorthand using the nearest **noble gas** in square brackets:

$$[Ar]\,3d^6\,4s^2$$

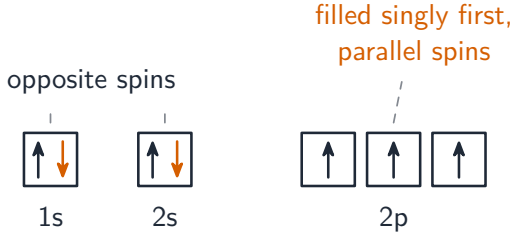
Here [Ar] stands for the full configuration of argon. Chromium and copper are exceptions to mechanically placing two electrons in 4s: Cr is $[Ar]3d^5 4s^1$ and Cu is $[Ar]3d^{10}4s^1$. Both totals and ground-state occupations matter.

For ions, you add or remove electrons. One key rule: when a transition metal forms a positive ion, it loses its 4s electrons **before** its 3d electrons. So Fe^{3+} is $[Ar]\,3d^5$.

Electrons in boxes

The **electrons in boxes** notation draws each orbital as a box and each electron as an arrow. Two electrons in the same orbital must point opposite ways, because each electron has a property called **spin** , and a shared orbital needs opposite spins.

Within a sub-shell, electrons fill empty orbitals one at a time, with parallel arrows, before any orbital gets a second electron. Spreading out like this keeps the electrons apart and lowers the repulsion between them.



Electrons in boxes for nitrogen ($1s^2\,2s^2\,2p^3$): paired electrons point opposite ways, and the 2p orbitals fill singly with parallel spins

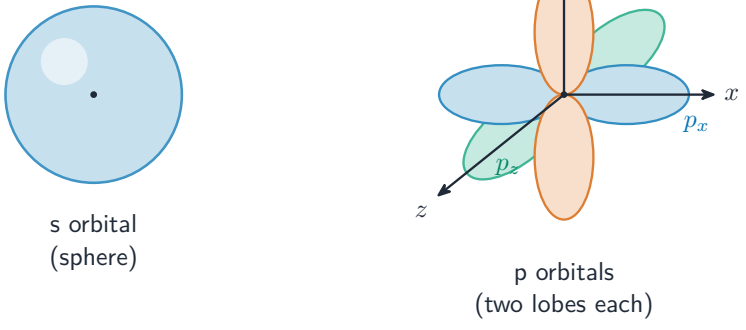
For oxygen, $2p^4$ has one pair and two singly occupied orbitals. The n = 2 shell also contains the 2s pair, so it contains two pairs altogether. A whole-shell question and a named-sub-shell question have different counting boundaries.

Why the configuration takes this shape

Electrons fill from low energy to high energy because that gives the most stable (lowest-energy) atom. Within a sub-shell they spread out singly first to reduce the repulsion between the negative electrons.

Shapes of s and p orbitals

- an s orbital is a **sphere** centred on the nucleus.
- a p orbital has two lobes, like a dumbbell, pointing along one axis. The three p orbitals point along three directions at right angles (the x, y and z axes).



An s orbital is a sphere; each p orbital is a dumbbell, and the three p orbitals point along the x, y and z axes

Free radicals

A **free radical** is a species with one or more **unpaired electrons**. Many free radicals are reactive, but the defining feature is an unpaired electron, not reactivity alone or a net charge.

Ionisation energy

First ionisation energy

The **first ionisation energy** (IE) is the energy needed to remove one electron from each atom in one mole of gaseous atoms, forming one mole of gaseous +1 ions.

Starting from separate gaseous atoms isolates electron removal without adding energy for breaking a solid or molecule into atoms. The unit is kJ mol^{-1}. As an equation, for an element X:

$$X(g) \rightarrow X^{+}(g) + e^{-}$$

The (g) shows the species is a gas.

Successive ionisation energies

After removing one electron, you can remove another. The second ionisation energy removes one electron from each +1 ion:

$$X^+(g) \rightarrow X^{2+}(g) + e^-$$

You can keep going. These are the **successive ionisation energies** . Each is larger than the one before, because every electron is pulled away from a more positive ion.

What ionisation energy depends on

Ionisation energy comes from the attraction between the positive nucleus and the outer electron. Three main factors set how strong that attraction is:

- **nuclear charge**: more protons pull the electrons more strongly, so the ionisation energy is higher.
- **atomic radius**: the further the outer electron sits from the nucleus, the weaker the pull, so the ionisation energy is lower.
- **shielding**: inner shells block some of the pull on the outer electron. More inner shells mean more shielding and a lower ionisation energy.

There is a smaller effect too – **spin-pair repulsion** . When two electrons share one orbital, they push each other a little, so one is easier to remove.

Trends in first ionisation energy

Across a period, the first ionisation energy generally **rises**. The nuclear charge grows while shielding stays about the same, so the outer electrons are held more tightly.

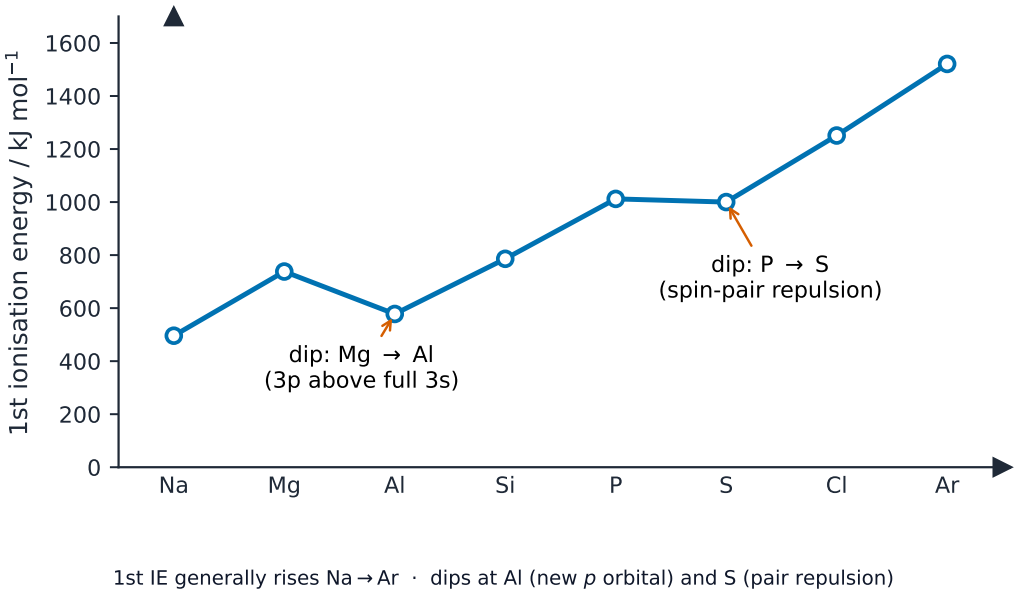
Down a group, the first ionisation energy **falls**. Lower elements have more shells, so more shielding and a larger radius, and the outer electron is easier to remove.

The dips are evidence for sub-shells

The rise across a period is not smooth. Two small dips appear, and you should be able to explain both:

- **Group 2 to Group 13** (for example Mg to Al): the electron removed from Al comes from a 3p sub-shell, which is higher in energy than the full 3s sub-shell in Mg. A 3p electron is easier to remove, so the value dips.
- **Group 15 to Group 16** (for example P to S): in S, one 3p orbital now holds a pair of electrons. Spin-pair repulsion makes one of them easier to remove, so the value dips.

These dips are evidence that sub-shells exist.



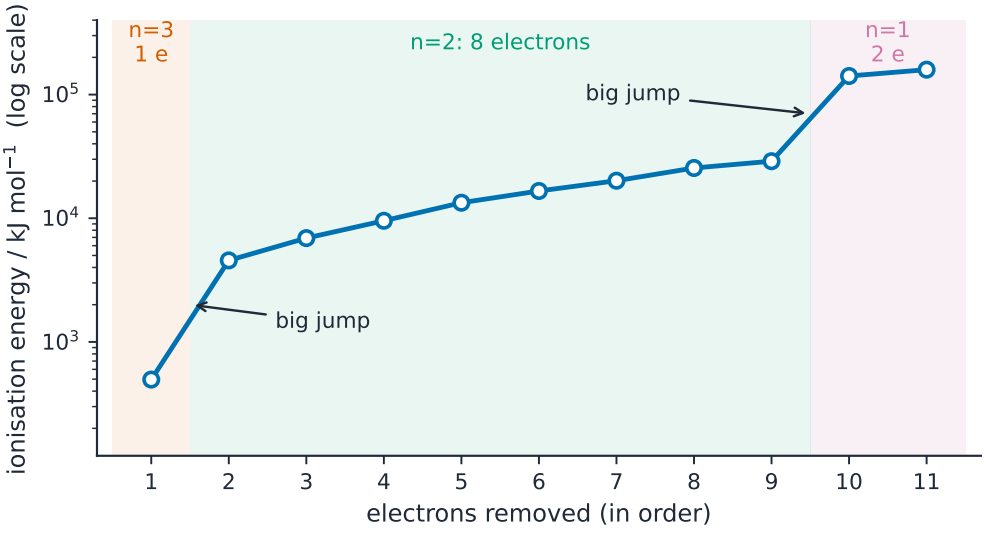
1st IE generally rises Na → Ar · dips at Al (new *p* orbital) and S (pair repulsion)

First ionisation energy rises across Period 3 but dips at Al and at S – evidence that sub-shells exist

■ Successive ionisation energies are evidence for shells

If you plot the successive ionisation energies of one element, the values rise, with **big jumps** at certain points. A big jump happens when the next electron must come from a shell closer to the nucleus.

For main-group atoms, count the electrons removed before the first inner-shell jump to find the outer-shell count. Counts 1 and 2 correspond to Groups 1 and 2; counts 3–8 correspond to Groups 13–18, with helium as the two-electron Group 18 exception. The first jump alone does not establish the period: use the number of occupied shells from fuller data or additional information about the atom.



big jump = next electron from a shell closer to the nucleus (shell structure)

Successive ionisation energies of sodium (log scale): the big jumps reveal the $2, 8, 1$ *shell structure*

Electrons removed before the first big jump	Group
1	Group 1
2	Group 2
3	Group 13

Worked example. The first five successive ionisation energies of an element are 590, 1150, 4940, 6480 and 8120 $\mathrm{kJ\,mol^{-1}}$. Which group is it in? Look for the **big jump**, not the biggest number. From the 1st to the 2nd the value roughly doubles, which is a normal rise. From the 2nd (1150) to the 3rd (4940) it more than quadruples: that is the jump. So **two** electrons come off easily before it, the outer shell holds 2 electrons, and the element is in **Group 2**. Count the electrons removed **before** the jump, and explain the jump properly: the next electron is being pulled from a shell **closer to the nucleus**, not from a different element.

Exam tips

- Define **isotopes** in full: atoms of the same element with the same number of protons but a different number of neutrons – the mark scheme wants both halves.
- 4s fills **before** 3d, but electrons are **removed from 4s first** when forming ions, so Fe^{3+} is $[Ar]3d^5$ (not $[Ar]3d^34s^2$).
- In successive ionisation energies, a **big jump** marks the start of a new (inner) shell – use the jumps to place the element in its group.

- Explain every ionisation-energy trend with the same three factors: **nuclear charge, distance and shielding**, plus sub-shell effects for the small dips.
- Learn the exact reason first ionisation energy of oxygen is **below** nitrogen: oxygen's **paired 2p electrons repel**, so one is easier to remove.

Atoms, molecules and stoichiometry

A-Level Chemistry

Relative masses of atoms and molecules

Atoms are far too light to weigh in grams, so we compare every mass to one standard. The standard is the **unified atomic mass unit** (symbol u), defined as exactly one twelfth of the mass of one carbon-12 atom.

Using this unit, we state masses as simple numbers:

- the **relative atomic mass** A_r of an element is the average mass of its atoms compared with $\frac{1}{12}$ of a carbon-12 atom. It is an average over all the **isotopes**, weighted by how common each one is.
- the **relative isotopic mass** is the mass of one atom of a single isotope, compared with $\frac{1}{12}$ of a carbon-12 atom.
- the **relative molecular mass** M_r of a **molecule** is the sum of the relative atomic masses of all its atoms.
- the **relative formula mass** is the same idea for a substance that is not made of molecules (such as an ionic compound). Add up the relative atomic masses shown in the formula.



$$A_r = \frac{(75 \times 35) + (25 \times 37)}{100} = 35.5$$

Relative atomic mass is a weighted average: chlorine's two isotopes (75% ^{35}Cl, 25% ^{37}Cl) average to $A_r = 35.5$

To find A_r from isotope data, multiply each isotope mass by its percentage, add these up, and divide by 100.

> **Worked example.** Chlorine is 75% ^{35}Cl and 25% ^{37}Cl. Find its relative atomic mass.
>
> $$A_r = \frac{(75 \times 35) + (25 \times 37)}{100} = \frac{2625 + 925}{100} = 35.5.$$

> **Worked example.** A mass spectrometer shows copper is 69.2% ^{63}Cu and 30.8% ^{65}Cu. Find its relative atomic mass.
>
> $$A_r = \frac{(69.2 \times 63) + (30.8 \times 65)}{100} = \frac{4359.6 + 2002.0}{100} = 63.6.$$

The mole and the Avogadro constant



A modern electronic balance measures mass – the basis of mole calculations.

Chemists count particles in groups called moles, just as we count eggs in dozens.

One **mole** (symbol mol) is the amount of substance that contains the same number of particles as there are atoms in exactly 12 g of carbon-12. That number is the **Avogadro constant**:

$$N_A = 6.02 \times 10^{23}\ \text{mol}^{-1}$$

So one mole of anything contains 6.02×10^{23} particles. The particles may be atoms, molecules or **ions** – always say which.

The mass of one mole in grams equals the relative mass (A_r or M_r). This is the **molar mass**, with units g mol^{-1}. The key equation is:

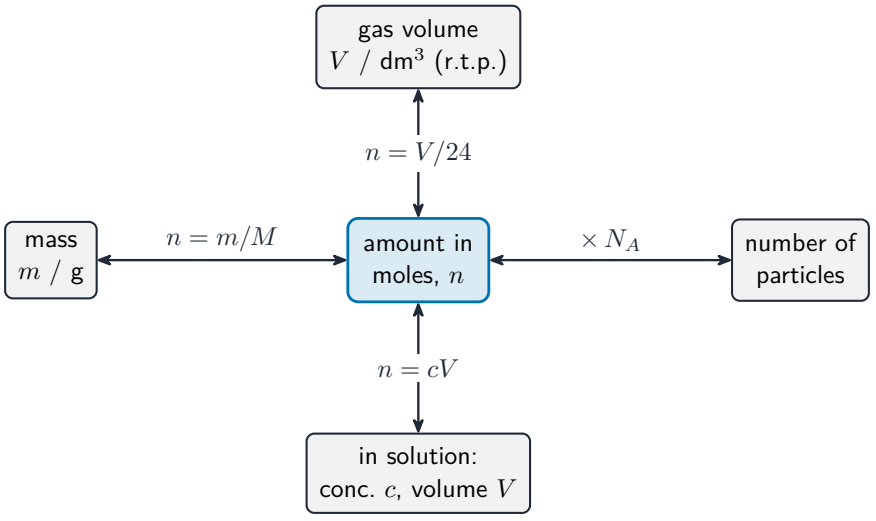
$$n = \frac{m}{M}$$

where n is the amount in moles, m is the mass in grams, and M is the molar mass.

> **Worked example.** How many moles are in 8.0 g of methane, CH_4? (A_r: C = 12, H = 1.)

The molar mass is $M = 12 + 4(1) = 16$ g mol^{-1}, so

$$n = \frac{m}{M} = \frac{8.0}{16} = 0.50 \text{ mol}.$$



The mole is the hub of every amount calculation: convert to mass ($n = m/M$), particles ($\times N_A$), gas volume ($n = V/24$) or solution ($n = cV$)

Formulas

A **compound** is a substance made of two or more elements chemically joined.

Charges and formulas of ionic compounds

In an **ionic compound** the total positive charge balances the total negative charge, so the compound is neutral overall.

You can predict the charge of many ions from the element's position in the Periodic Table:

Group	**1**	**2**	**13**	**15**	**16**	**17**
Usual ion charge	+1	+2	+3	−3	−2	−1

Hydrogen forms H^+, and the Group 18 noble gases do not normally form ions.

Some ions you must know by name and formula:

Name	**Formula**
nitrate	NO_3^-
carbonate	CO_3^{2-}
sulfate	SO_4^{2-}
hydroxide	OH^-
ammonium	NH_4^+
zinc	Zn^{2+}
silver	Ag^+
hydrogencarbonate	HCO_3^-

Name	Formula
phosphate	PO_4^{3-}

For a metal that can have more than one charge, a Roman numeral shows the **oxidation number**. For example, iron(II) is Fe^{2+} and iron(III) is Fe^{3+}. To write a formula, balance the charges: iron(III) oxide is Fe_2O_3, because two Fe^{3+} balance three O^{2-}.

Equations and state symbols

A chemical equation must be **balanced** – the same number of each kind of atom on both sides. Add **state symbols** to show the state of each species: (s) solid, (l) liquid, (g) gas, and (aq) **aqueous** (dissolved in water).

An **ionic equation** shows only the ions and molecules that actually change. The ions that do not change are **spectator ions**, and you leave them out. For example, the reaction that forms silver chloride is:

$$Ag^+(aq) + Cl^-(aq) \rightarrow AgCl(s)$$

Empirical and molecular formulas

The **empirical formula** is the simplest whole-number ratio of the atoms of each element in a compound. The **molecular formula** shows the actual number of atoms of each element in one molecule.

For example, ethane has empirical formula CH_3 but molecular formula C_2H_6.

Hydrated and anhydrous solids

Some solids hold water inside their crystals. This water is the **water of crystallisation**. A solid that contains it is **hydrated**; the same solid with the water removed is **anhydrous**.

For example, hydrated copper(II) sulfate is $CuSO_4 \cdot 5H_2O$. Heating it drives off the water to leave anhydrous $CuSO_4$.

Calculating empirical and molecular formulas

To find the empirical formula from masses (or percentages by mass):

1. divide each element's mass by its A_r to get the moles.
2. divide all the mole values by the smallest one.
3. round to the nearest whole numbers – that ratio is the empirical formula.

To get the molecular formula, you also need M_r. Find how many times the empirical formula mass fits into M_r, then multiply the formula by that number.

Worked example. A compound is 40.0% carbon, 6.7% hydrogen and 53.3% oxygen by mass. Find its empirical formula. (A_r: C = 12, H = 1, O = 16.)

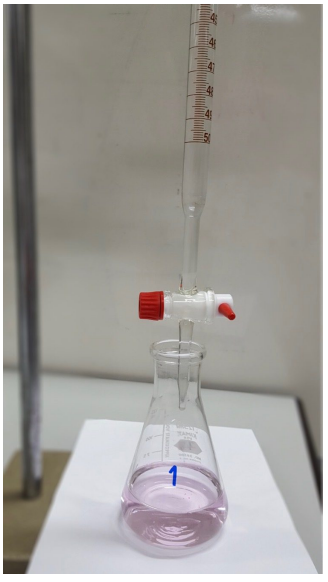
Take 100 g and divide each mass by its A_r to get moles: C = 40.0/12 = 3.33, H = 6.7/1 = 6.7, O = 53.3/16 = 3.33. Dividing through by the smallest (3.33) gives a ratio C : H : O = 1 : 2 : 1, so the empirical formula is CH_2O.

element	% by mass	÷ A_r → moles	÷ smallest (3.33)	ratio
C	40.0	40.0/12 = 3.33	1.0	1
H	6.7	6.7/1 = 6.7	2.0	2
O	53.3	53.3/16 = 3.33	1.0	1

empirical formula CH_2O - if $M_r = 180$, it fits $180/30 = 6$ times: molecular formula $C_6H_{12}O_6$

The empirical-formula recipe: divide by Ar, divide by the smallest, read off the ratio

Reacting masses and volumes



A titration finds reacting volumes precisely.

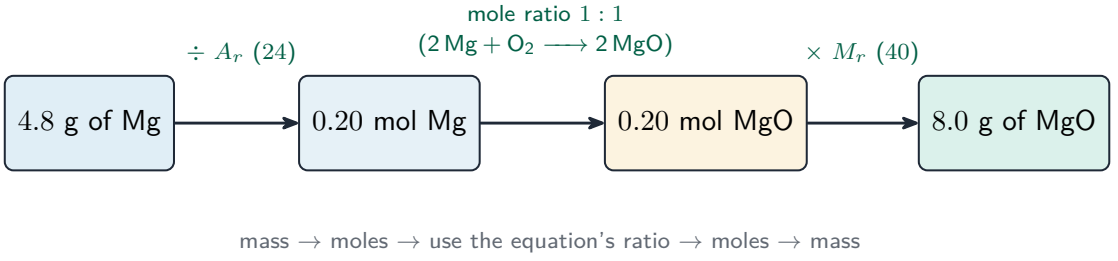
■ Reacting masses and percentage yield

The numbers in front of each species in a balanced equation give the mole ratio – this is the **stoichiometry** . To find a reacting mass: change the known mass to moles, use the mole ratio to find the moles you want, then change back to mass.

Worked example. What mass of magnesium oxide forms when 4.8 g of magnesium burns completely? $2\text{Mg} + \text{O}_2 \rightarrow 2\text{MgO}$. ($A_r$: Mg = 24, O = 16.)

Moles of Mg = $4.8/24 = 0.20$ mol. The ratio Mg : MgO is 1 : 1, so 0.20 mol of MgO forms. Its molar mass is $24 + 16 = 40$ g mol^{-1}, so

$$m = nM = 0.20 \times 40 = 8.0 \text{ g}.$$



The mole bridge: mass to moles, ratio, then back to mass

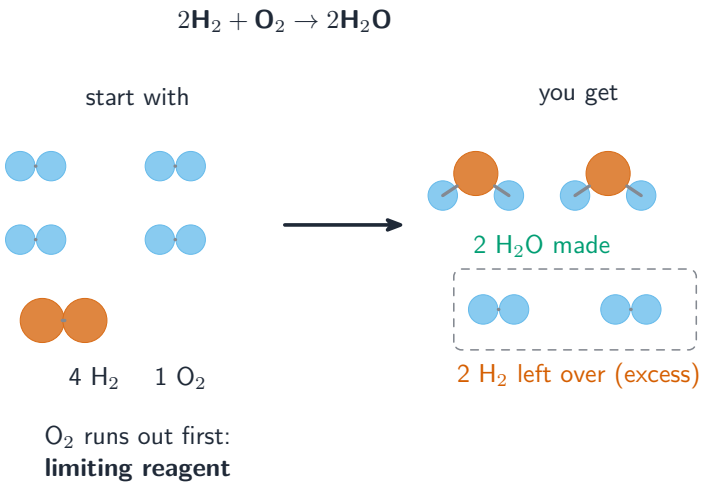
In real reactions you usually get less product than the maximum. The **percentage yield** compares the amount you actually made with the most you could make:

$$\text{percentage yield} = \frac{\text{actual amount of product}}{\text{maximum possible amount}} \times 100\%$$

Limiting and excess reagent

When two reactants are mixed, one usually runs out first. The **limiting reagent** is the one that runs out – it decides how much product forms. The other is the **excess reagent**, because there is more than enough of it. Always base the calculation on the limiting reagent.

To find it: work out the moles of each reactant, divide each by its number in the equation, and the smallest result is the limiting reagent.

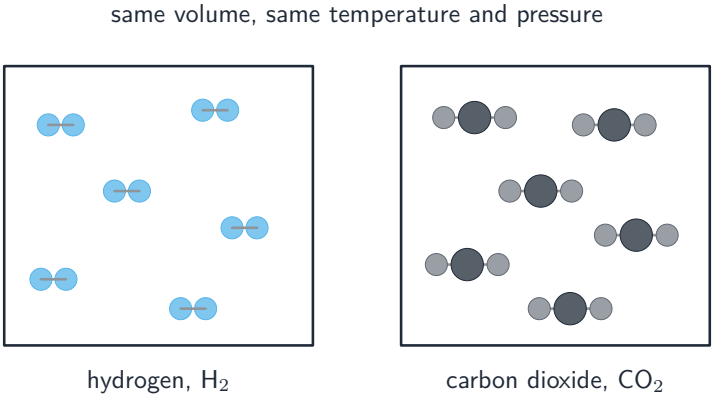


The limiting reagent runs out first and decides how much product forms; the leftover reactant is in excess

Volumes of gases

At the same temperature and pressure, equal volumes of any gases contain equal numbers of molecules. At room temperature and pressure (r.t.p.), one mole of any gas takes up 24.0 dm^3, so:

$$n = \frac{V}{24.0} \qquad (V \text{ in } dm^3 \text{ at r.t.p.})$$



Equal volumes of gases at the same temperature and pressure hold equal numbers of molecules, whatever the gas

This is used when burning **hydrocarbons** (compounds of only carbon and hydrogen), where you compare gas volumes.

Volumes and concentrations of solutions

The **concentration** of a solution is the amount of **solute** in each cubic decimetre of **solution**, measured in mol dm^{-3}:

$$n = c \times V$$

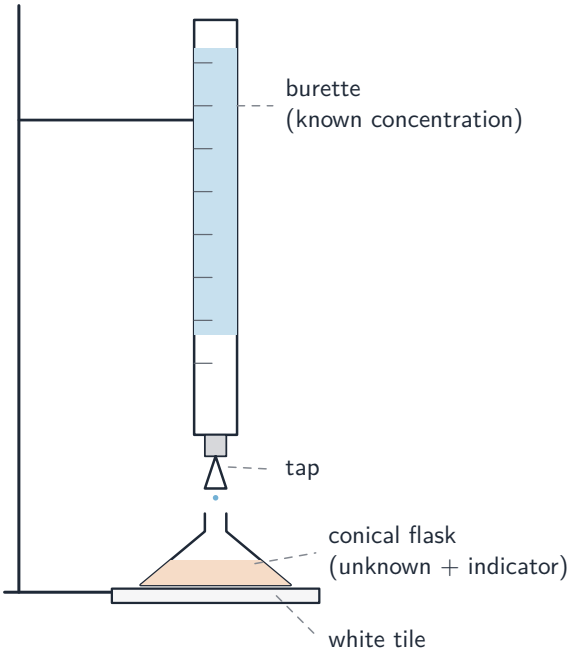
where c is the concentration and V is the volume in dm^3. Remember that 1000 cm^3 = 1 dm^3.

> **Worked example.** In a titration, 25.0 cm^3 of sodium hydroxide solution is exactly neutralised by 20.0 cm^3 of 0.100 mol dm^{-3} hydrochloric acid: $NaOH + HCl \rightarrow NaCl + H_2O$. Find the concentration of the sodium hydroxide.

Moles of HCl $= cV = 0.100 \times \dfrac{20.0}{1000} = 2.00 \times 10^{-3}$ mol. The ratio is 1 : 1, so there are 2.00×10^{-3} mol of NaOH in 25.0 cm^3:

$$c = \frac{n}{V} = \frac{2.00 \times 10^{-3}}{25.0/1000} = 0.0800 \text{ mol dm}^{-3}.$$

This is the basis of a **titration**, where you find an unknown concentration by reacting it with a solution whose concentration you already know.



In a titration a burette adds a solution of known concentration to the unknown in the conical flask, until the indicator changes

■ Significant figures

Give your answer to a sensible number of **significant figures** – usually match the data in the question. Do not write more digits than the data supports, and do not round so early that you lose accuracy.

Exam tips

- Convert volumes to dm^3 **before** using concentration (25.0 cm^3 = 0.0250 dm^3); the missing ÷1000 is the most common titration error.
- Use the **balancing numbers** as the mole ratio between species – never the M_r values.
- Empirical formula: divide each mass/percentage by A_r, then by the **smallest**, then scale to whole numbers; use M_r to reach the molecular formula.
- For gases at r.t.p. use volume = moles × 24 dm^3; quote the equation you use every time.
- Give the answer to the same **significant figures** as the data (usually 3) and always include units.

Chemical bonding

A-Level Chemistry

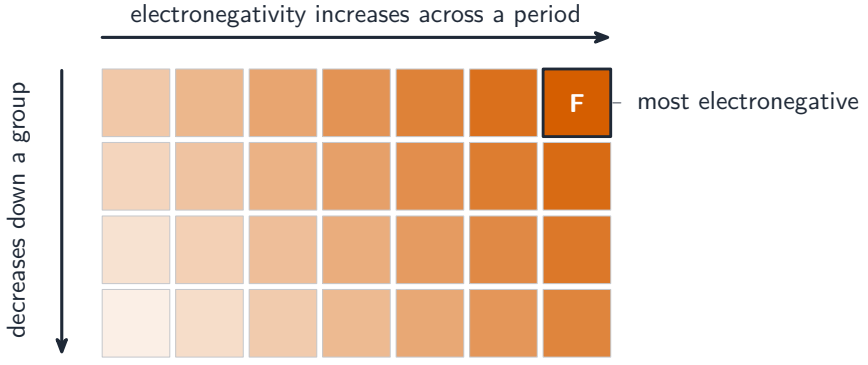
Electronegativity

Electronegativity is the power of an atom to attract the **electrons** in a bond towards itself.

Three factors decide how electronegative an atom is:

- **nuclear charge**: more protons pull the bonding electrons more strongly.
- **atomic radius**: the closer the bond is to the nucleus, the stronger the pull.
- **shielding** by inner shells and sub-shells: more inner electrons weaken the pull on the bonding electrons.

So electronegativity **rises** across a period (more nuclear charge, smaller radius) and **falls** down a group (larger radius, more shielding). Fluorine is the most electronegative element.



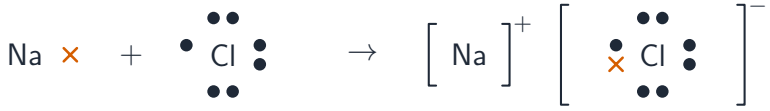
Electronegativity rises across a period and falls down a group, so fluorine is the most electronegative element

You can use the difference in **Pauling electronegativity** values to predict the bond type. A large difference gives an **ionic** bond; a small difference gives a **covalent** bond.

Ionic bonding

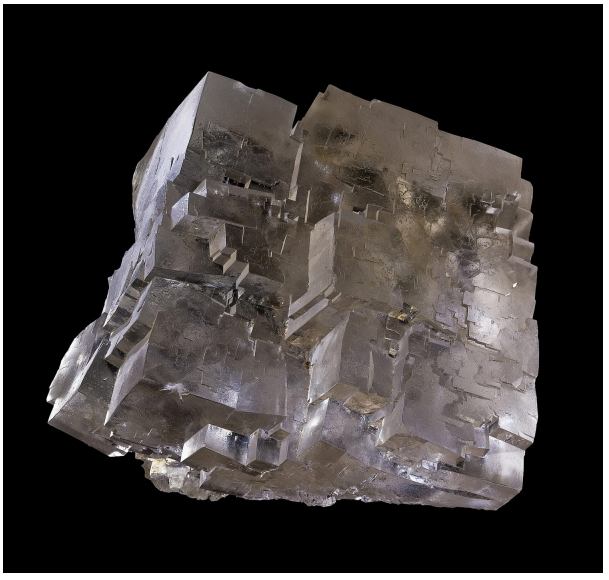
Ionic bonding is the **electrostatic attraction** between oppositely charged **ions** – positive **cations** and negative **anions**.

It forms when a metal gives electrons to a non-metal. Good examples are sodium chloride (NaCl), magnesium oxide (MgO) and calcium fluoride (CaF_2). The ions pack into a regular giant **lattice**, held together by the attraction in every direction.



sodium gives its outer electron (×) to chlorine

Ionic bonding in NaCl: sodium transfers its single outer electron to chlorine, giving Na^+ and a full-octet Cl^-

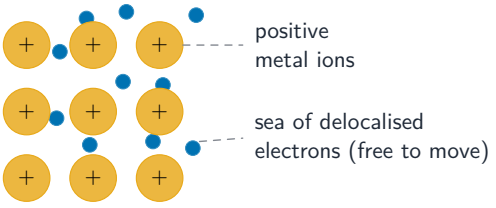


A real crystal of rock salt (halite, NaCl); the cubic shapes mirror the giant ionic lattice inside

Metallic bonding

Metallic bonding is the electrostatic attraction between positive metal ions and a "sea" of **delocalised electrons** .

The outer electrons are free to move through the whole metal. This explains why metals conduct electricity and are strong.



Metallic bonding: positive metal ions sit in a sea of delocalised electrons that are free to move

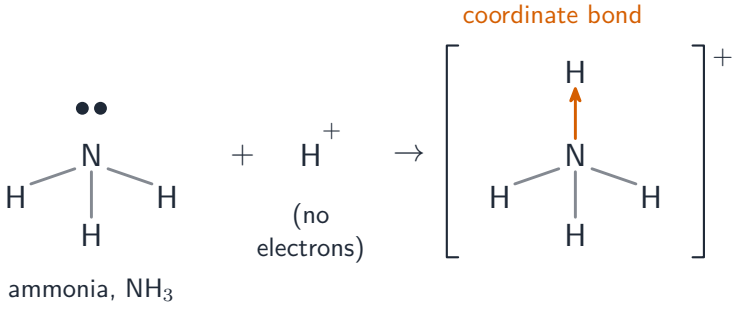
Covalent and coordinate bonding

Covalent bonding is the electrostatic attraction between the nuclei of two atoms and a shared pair of electrons.

Simple molecules with covalent bonds include H_2, O_2, N_2, Cl_2, HCl, CO_2, NH_3, CH_4, C_2H_6 and C_2H_4. A double bond shares two pairs; a triple bond (as in N_2) shares three pairs.

Atoms in Period 3 and below can **expand the octet** – hold more than eight electrons in their outer shell. Examples are SO_2, PCl_5 and SF_6.

A **coordinate bond** (also called a dative covalent bond) is a covalent bond where both shared electrons come from the same atom. For example, when ammonia and hydrogen chloride gases meet, the lone pair on the nitrogen forms a coordinate bond to H^+, making the ammonium ion NH_4^+. Coordinate bonds also join the two halves of the Al_2Cl_6 molecule.



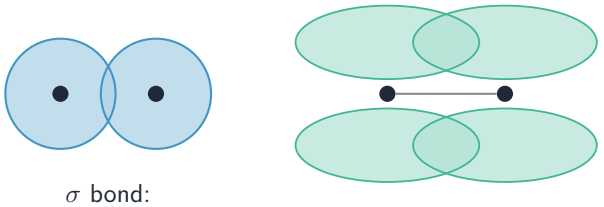
A coordinate (dative) bond: nitrogen's lone pair forms the fourth N–H bond, both electrons coming from N

Sigma and pi bonds

Covalent bonds form when **orbitals** overlap:

- a **sigma bond** σ forms by the direct, head-on **overlap** of orbitals between the two atoms.
- a **pi bond** π forms by the sideways overlap of two p orbitals, above and below the sigma bond.

A single bond is one sigma bond. A double bond (as in C_2H_4) is one sigma plus one pi bond. A triple bond (as in N_2 and HCN) is one sigma plus two pi bonds.



A σ bond forms by direct head-on overlap; a π bond forms by the sideways overlap of two p orbitals, above and below

Hybridisation

Hybridisation mixes orbitals in the same shell to make new, equal orbitals for bonding:

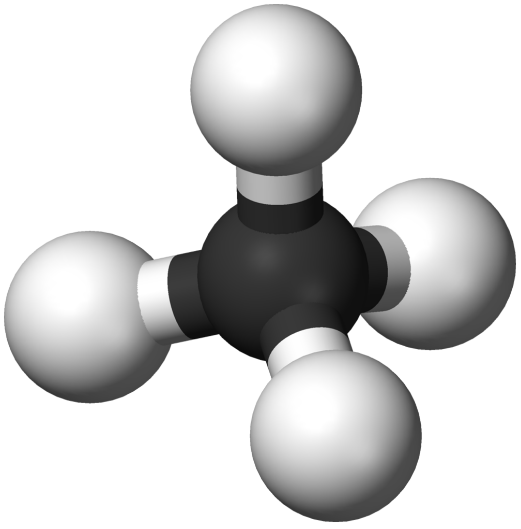
- sp: two equal orbitals, used in a linear molecule.
- sp^2: three equal orbitals, used in a flat molecule like C_2H_4.
- sp^3: four equal orbitals, used in CH_4.

Bond energy and bond length

- **bond energy** is the energy needed to break one mole of a particular covalent bond in the gas state.
- **bond length** is the distance between the centres of the two bonded atoms.

A shorter bond is usually stronger (higher bond energy). Triple bonds are shorter and stronger than double bonds, which are shorter and stronger than single bonds. Stronger bonds make a molecule harder to react.

Shapes of molecules



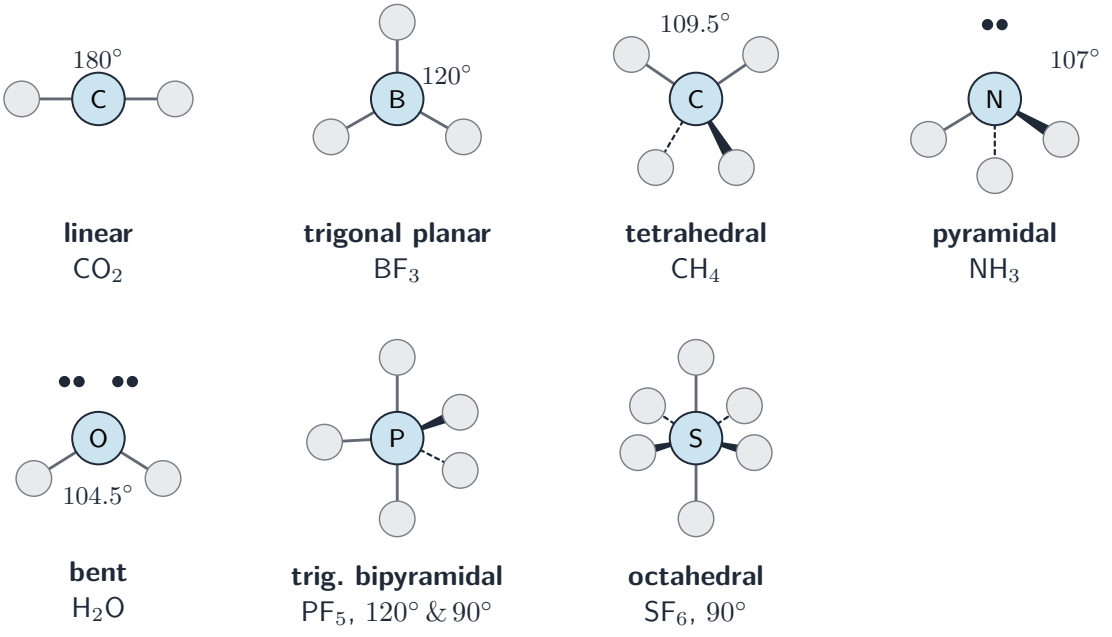
A molecular model shows the three-dimensional shape of a covalent molecule.

To work out a shape, use **VSEPR theory** : the pairs of electrons around the central atom push apart as far as possible, because like charges repel.

A **lone pair** (not in a bond) pushes more strongly than a **bonding pair** . Each lone pair squeezes the **bond angle** by about 2.5°.

Molecule	Shape	Bond angle
CO_2	**linear**	180°
BF_3	**trigonal planar**	120°
CH_4	**tetrahedral**	109.5°
NH_3	**pyramidal**	107°
H_2O	**bent**	104.5°
PF_5	**trigonal bipyramidal**	120° and 90°
SF_6	**octahedral**	90°

NH_3 has one lone pair and H_2O has two, which is why their angles drop below the 109.5° of CH_4. You can predict the shapes of similar molecules and ions in the same way.



The seven shapes from VSEPR theory; the lone pairs on NH_3 and H_2O push harder, squeezing the bond angle below 109.5°

Worked example. Predict the shape and bond angle of NH_3 and of H_2O. Nitrogen has 5 outer electrons and forms 3 bonds, leaving **3 bonding pairs and 1 lone pair** - four pairs in total, so they start from the tetrahedral 109.5°. The lone pair repels more strongly and squeezes the angle by about 2.5°, giving a **pyramidal** shape at about 107°. Oxygen forms 2 bonds and keeps **2 lone pairs**: still four pairs, but now two squeezes, so the shape is **bent** at about 104.5°. Count **all** the pairs to fix the basic geometry, subtract 2.5° for **each** lone pair, and name the shape from the **atoms** - NH_3 has four pairs but is pyramidal, not tetrahedral.

Intermolecular forces

Intermolecular forces are the forces *between* **molecules**. They are much weaker than the ionic, covalent and metallic bonding *inside* substances.

■ Bond polarity and dipoles

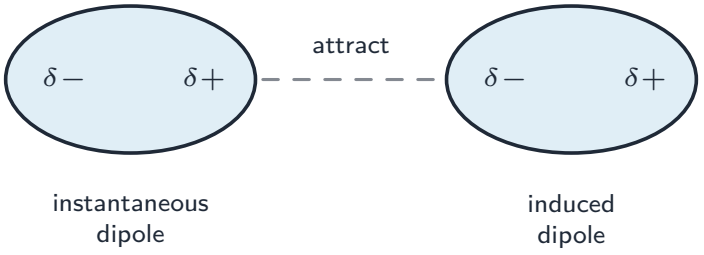
When two atoms with different electronegativity share a bond, the electrons sit closer to the more electronegative atom. The bond then has a **polarity**: one end is slightly negative ($\delta-$) and the other slightly positive ($\delta+$). This separation of charge is a **dipole**.

If the dipoles in a molecule do not cancel, the whole molecule has a **dipole moment** and is polar. If they cancel by symmetry (as in CO_2), the molecule is non-polar.

Van der Waals' forces

Van der Waals' forces is the general name for all intermolecular forces. There are two main types.

The first type is the instantaneous dipole–induced dipole force, also called the **London dispersion force** . Moving electrons make a brief **instantaneous dipole** , which then creates a matching **induced dipole** in a nearby molecule. These forces act between all molecules and get stronger when there are more electrons.

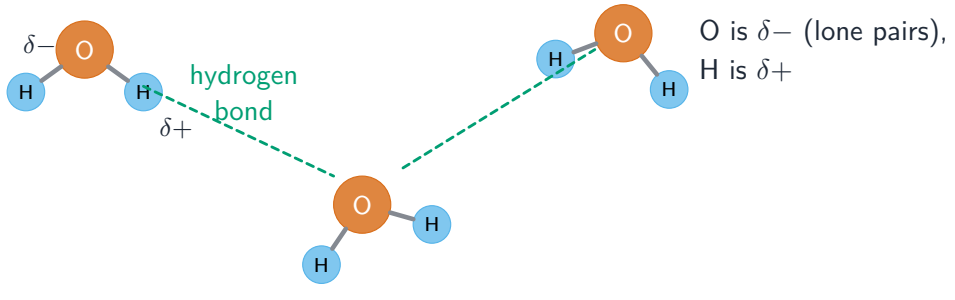


A London force: a momentary dipole in one molecule induces a dipole in its neighbour, so they attract – this acts between all molecules

The second type is the permanent dipole–permanent dipole force. It acts between molecules that are always polar, because each one has a **permanent dipole** .

Hydrogen bonding

Hydrogen bonding is a strong, special case of permanent dipole forces. It forms when hydrogen is bonded to a very electronegative atom – nitrogen, oxygen or fluorine – and is attracted to a lone pair on an N, O or F atom in a neighbour. Look for N–H and O–H groups, as in ammonia and water.



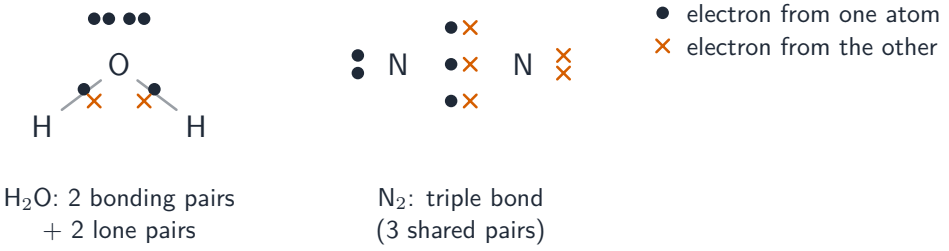
Hydrogen bonding in water: a δ+ hydrogen is attracted to a lone pair on the δ− oxygen of a neighbouring molecule

Hydrogen bonding explains the strange behaviour of water:

- its high melting and **boiling point** , because many hydrogen bonds must be broken.
- its high **surface tension** .
- ice is **less dense** than liquid water, because hydrogen bonds hold the molecules in an open, spread-out structure, so ice floats.

Dot-and-cross diagrams

A **dot-and-cross diagram** shows the outer electrons of each atom, using dots for one atom and crosses for the other. This makes it clear where each bonding electron came from. You can draw them for ionic, covalent and coordinate bonding, including molecules with an expanded octet or an odd number of electrons.



Covalent dot-and-cross: each shared pair is one electron from each atom. Water has two bonding pairs and two lone pairs; nitrogen shares three pairs (a triple bond)

Exam tips

- For **shapes**, count bonding pairs and lone pairs, name the shape, then give the exact bond angle (e.g. NH_3: pyramidal, 107°) – each lone pair lowers the angle by about 2.5°.
- A **dative (coordinate) bond** has both electrons from one atom (e.g. NH_4^+, H_3O^+); draw the arrow from the lone pair.
- Name the intermolecular force precisely: **hydrogen bonding** needs H bonded to N, O or F; otherwise it is permanent-dipole or induced-dipole (van der Waals). Never call van der Waals forces "bonds".
- Explain a physical property by stating **which forces are broken**, not just "strong bonds".

States of matter

A-Level Chemistry

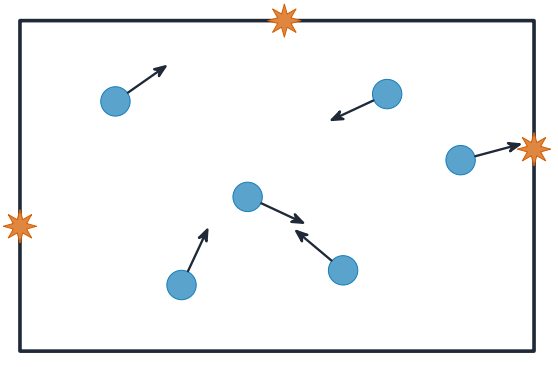
The gaseous state



Boiling turns liquid water into steam – a change between states of matter.

Where gas pressure comes from

Gas molecules move fast in all directions. They keep hitting – **colliding** with – the walls of their container. Each hit gives the wall a tiny push. The **pressure** of the gas is the overall result of these many collisions on the walls.



fast particles collide with the walls; the many pushes make the **pressure**

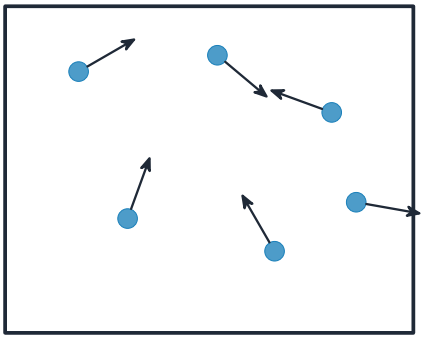
Gas pressure: fast molecules move in all directions and collide with the walls; the many tiny pushes add up to the pressure

Ideal gases

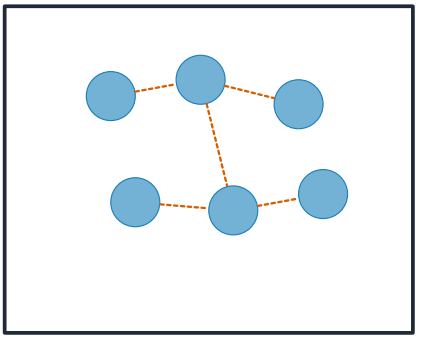
An **ideal gas** is a simple model. We assume two things:

- the particles themselves take up zero volume.
- there are no **intermolecular forces** of attraction between the particles.

A **real gas** follows this model closely at low pressure and high temperature. It behaves least like an ideal gas at high pressure and low temperature, when the particles are squeezed close together and the forces between them start to matter.



ideal gas (model)
point particles, *no* forces



real gas: high p, low T
real size + attractions (dashed)

An ideal gas is a model: point particles with no forces between them. A real gas behaves least like this at high pressure and low temperature, when the particles are crowded and their real size and attractions start to matter

The ideal gas equation

The **ideal gas equation** links pressure, volume, amount and temperature:

$$pV = nRT$$

where p is the pressure in Pa, V is the volume in m^3, n is the amount in moles, T is the temperature in **kelvin** (K), and R is the **gas constant** ($8.31\ \text{J K}^{-1}\,\text{mol}^{-1}$).

Always change the units first: °C to K (add 273), and cm^3 or dm^3 to m^3.

> **Worked example.** Find the volume of 0.50 mol of an ideal gas at 27 °C and 100 kPa. ($R = 8.31\ \text{J K}^{-1}\,\text{mol}^{-1}$.)

Convert first: $T = 300$ K, $p = 1.00 \times 10^5$ Pa. Then

$$V = \frac{nRT}{p} = \frac{0.50 \times 8.31 \times 300}{1.00 \times 10^5} = 0.0125\ \text{m}^3\ (= 12.5\ \text{dm}^3).$$

You can also use the equation to find a **molar mass** . Since $n = m/M$:

$$pV = \frac{m}{M}RT \qquad \Rightarrow \qquad M = \frac{mRT}{pV}$$

This lets you work out M_r from the mass (or the density) of a gas.

> **Worked example.** A flask holds 0.96 g of a gas in 600 cm^3 at 100 kPa and 27 °C. Find the molar mass of the gas. ($R = 8.31$ J K^{-1} mol^{-1}.)

Convert: $V = 6.00 \times 10^{-4}$ m^3, $T = 300$ K. Then

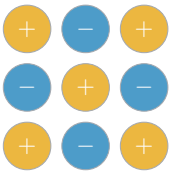
$$M = \frac{mRT}{pV} = \frac{0.96 \times 8.31 \times 300}{(1.00 \times 10^5)(6.00 \times 10^{-4})} \approx 40 \text{ g mol}^{-1}.$$

Bonding and structure

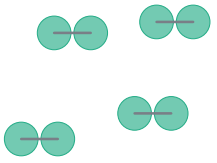


Quartz is a giant covalent structure of silicon and oxygen.

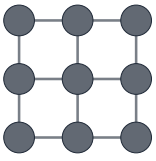
How a substance behaves depends on how its particles are joined. There are four main structures of a **crystalline solid** .



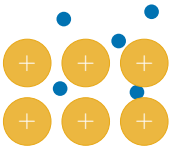
giant ionic
NaCl: high m.p.,
conducts when molten



simple molecular
I_2, ice: low m.p.,
weak forces between



giant covalent
diamond, SiO_2: very
high m.p., insulating



giant metallic
copper: high m.p.,
conducts (solid + molten)

The four structures of a crystalline solid – the structure decides the melting point, conductivity and solubility

Giant ionic

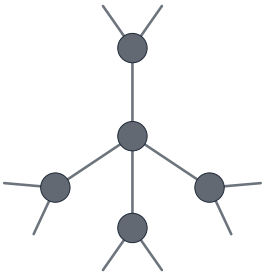
A **giant ionic** structure is a huge regular **lattice** of positive and negative **ions**, held together by strong attraction in every direction. Examples are sodium chloride and magnesium oxide.

Simple molecular

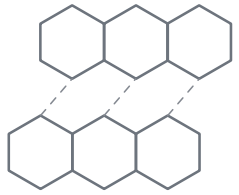
A **simple molecular** structure is made of small **molecules**. The bonds inside each molecule are strong, but the intermolecular forces between the molecules are weak. Examples are iodine (I_2), **fullerene** (C_{60}) and ice.

Giant molecular

A **giant molecular** structure (also called giant covalent) is a huge network of atoms joined by strong **covalent bonds**. Examples are **silicon(IV) oxide**, **graphite** and **diamond**.



diamond
every C bonded to 4:
rigid 3D network, very hard



graphite
layers slide; spare
electrons let it conduct

Two giant covalent forms of carbon: diamond is a rigid 3D network (very hard); graphite has sliding layers and spare electrons that conduct



A real diamond in the rock it grew in. That whole crystal is one giant molecule – a single unbroken network of carbon atoms, each bonded to four others. Breaking it means breaking countless strong covalent bonds, which is why diamond is the hardest natural material

■ Giant metallic

A **giant metallic** structure is a lattice of positive metal ions in a "sea" of **delocalised electrons**. An example is copper.



A piece of pure copper metal. The shine, and the way metals conduct and bend, all come from that giant lattice of copper ions sitting in a shared sea of delocalised electrons

Physical properties

The structure decides the physical properties:

Structure	Melting/boiling point	Conducts electricity?	Solubility in water
giant ionic	high	only when molten or dissolved	usually soluble
simple molecular	low	no	usually low
giant molecular	very high	no (except graphite)	insoluble
giant metallic	high	yes (solid and molten)	insoluble

- **melting point** and **boiling point** are high when strong forces (ionic, covalent or metallic) must be broken, and low when only weak intermolecular forces break.
- **electrical conductivity** needs charged particles that can move – ions that are free (when molten or dissolved) or delocalised electrons. Graphite conducts because some of its electrons are delocalised.
- **solubility** in water is usually high for ionic solids and low for molecular and giant covalent solids.

You can work backwards too: from the melting point, conductivity and solubility of an unknown substance, deduce the type of structure and bonding it has.

Exam tips

- Use **SI units** in $pV = nRT$: pressure in Pa, volume in m^3 ($cm^3 \times 10^{-6}$), temperature in K ($°C + 273$).
- State the **ideal-gas assumptions** (negligible molecular volume, no intermolecular forces) and when real gases deviate (high pressure, low temperature).
- Link each property to structure: giant ionic (high m.p., conducts molten), giant covalent (very high m.p.), simple molecular (low m.p.), giant metallic (conducts, malleable).
- Graphite conducts because each carbon has a **delocalised** electron; diamond does not – a favourite comparison.

Chemical energetics

A-Level Chemistry

Enthalpy change, ΔH



Burning fuel is exothermic, releasing energy to the surroundings.



An instant cold pack uses an endothermic reaction that takes in heat.

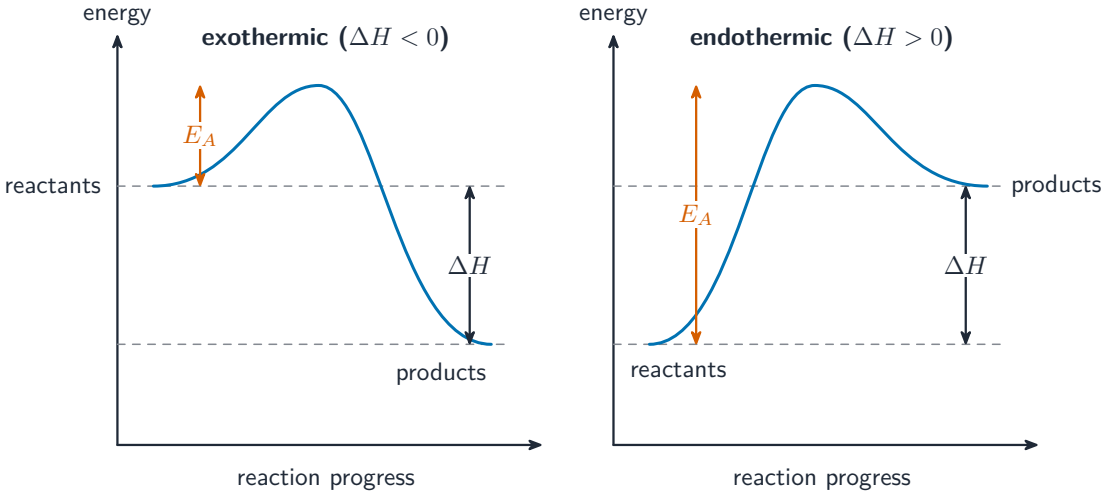
Every chemical reaction takes in or gives out energy. This energy change, measured at constant pressure, is the **enthalpy change** , with symbol ΔH.

- in an **exothermic** reaction the system gives out heat, so the products have less energy than the reactants and ΔH is **negative**.
- in an **endothermic** reaction the system takes in heat, so the products have more energy than the reactants and ΔH is **positive**.

Reaction pathway diagrams

A **reaction pathway diagram** shows the energy of the reactants and products, and the energy "hill" between them. The height of the hill is the **activation energy** – the least energy the particles need before they can react.

- exothermic: products sit **lower** than reactants ($\Delta H < 0$).
- endothermic: products sit **higher** than reactants ($\Delta H > 0$).

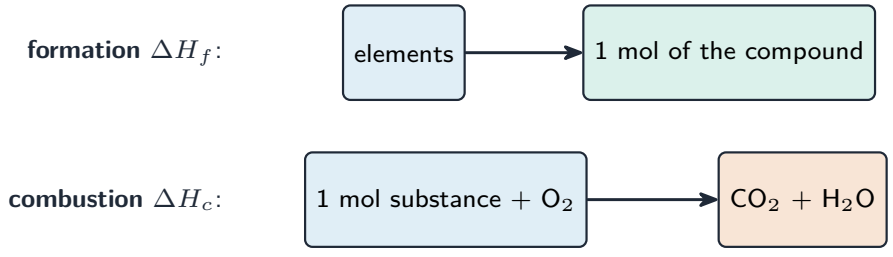


Exothermic reactions end lower than they start ($\Delta H < 0$); endothermic reactions end higher ($\Delta H > 0$)

Standard conditions and types of enthalpy change

Energy values are compared under **standard conditions**: 298 K and 101 kPa, shown by the symbol $^{\ominus}$. Each substance is in its normal physical state at those conditions.

Symbol	Name	Definition (per mole, under standard conditions)
$\Delta H_r^{\ominus}$	**enthalpy change of reaction**	for the amounts shown in the equation
$\Delta H_f^{\ominus}$	**enthalpy change of formation**	one mole of a compound forms from its elements
$\Delta H_c^{\ominus}$	**enthalpy change of combustion**	one mole of a substance burns completely in oxygen
$\Delta H_{\text{neut}}^{\ominus}$	**enthalpy change of neutralisation**	one mole of water forms from an acid and an alkali



Don't confuse them: formation builds 1 mol of a compound from its elements; combustion burns 1 mol of a substance in oxygen

Energy from breaking and making bonds

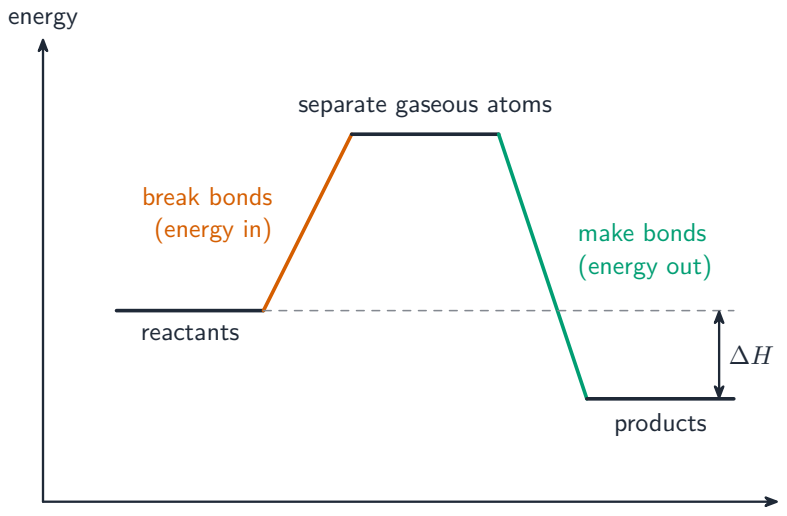
During a reaction, old bonds break and new bonds form. **Breaking** a bond needs energy (endothermic); **making** a bond releases energy (exothermic). The enthalpy change of the reaction is the difference between the two:

$$\Delta H_r = \sum(\text{bond energies broken}) - \sum(\text{bond energies made})$$

The **bond energy** is the energy needed to break one mole of a particular bond in the gas state, so it is always positive. Some bond energies are exact (for one specific molecule); others are averages taken over many different molecules, so calculations using them are only approximate.

> **Worked example.** Use bond energies to find ΔH for $H_2 + Cl_2 \rightarrow 2HCl$. Bond energies (kJ mol^{-1}): H–H = 436, Cl–Cl = 242, H–Cl = 431.

$$\Delta H = \sum(\text{broken}) - \sum(\text{made}) = (436 + 242) - (2 \times 431) = 678 - 862 = -184 \text{ kJ mol}^{-1}.$$



Breaking bonds takes energy in; making bonds gives energy out. ΔH is the difference between the two

Measuring enthalpy change in the lab

When a reaction heats up (or cools down) a known mass of water or solution, the heat transferred is:

$$q = mc\Delta T$$

where m is the mass, c is the **specific heat capacity** (how much energy raises 1 g by 1 K), and ΔT is the temperature change. The enthalpy change per mole is then:

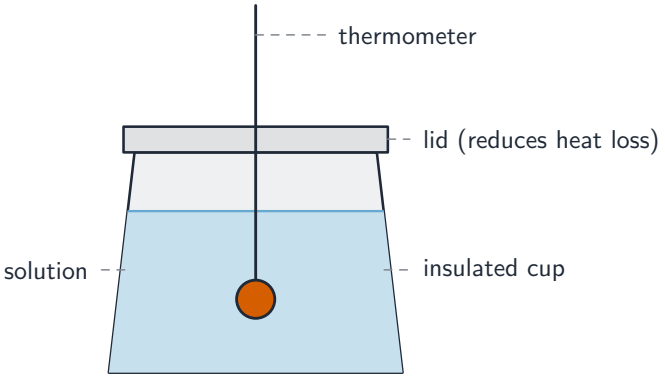
$$\Delta H = -\frac{mc\Delta T}{n}$$

The minus sign makes ΔH negative when the temperature rises (an exothermic reaction).

> **Worked example.** Burning 0.50 g of methanol ($M_r = 32$) raises the temperature of 100 g of water by 18 °C. Find the enthalpy change of combustion per mole. ($c = 4.18$ J g^{-1} K^{-1}.)

The heat released is $q = mc\Delta T = 100 \times 4.18 \times 18 = 7520$ J. The amount burnt is $n = 0.50/32 = 0.0156$ mol, so

$$\Delta H_c = -\frac{q}{n} = -\frac{7520}{0.0156} \approx -480 \text{ kJ mol}^{-1}.$$

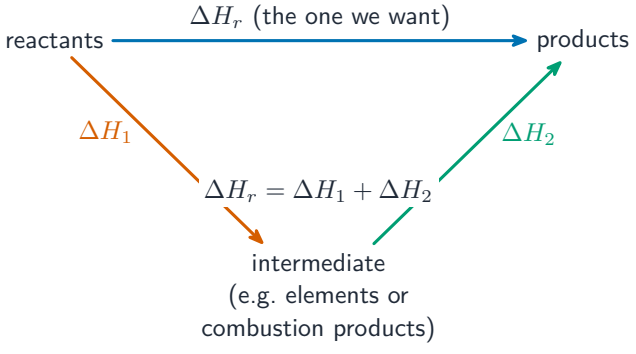


An insulated cup and a thermometer measure the temperature change of a known mass of solution

Hess's law

Hess's law says that the total enthalpy change for a reaction is the same, no matter which route you take from reactants to products. This is because energy is conserved.

This lets you draw an **energy cycle** : link the reactants and products by a direct step and by an indirect route, then add the steps so that both routes give the same total.



The direct route equals the indirect route, so $\Delta H_r = \Delta H_1 + \Delta H_2$

Hess's law is useful in two ways:

- it lets you find an enthalpy change that you **cannot** measure directly (for example, the formation of a compound that forms slowly or with side reactions).

- it lets you calculate ΔH_r from bond energy data, or from formation or combustion data given in the question.

Exam tips

- Define each enthalpy change with its **exact standard conditions** (e.g. combustion = one mole burned completely in excess oxygen).
- Use $\Delta H = \sum$(bonds broken) $- \sum$(bonds made); breaking is endothermic (+), making is exothermic (−) – getting the sign the wrong way round is the classic error.
- In $q = mc\Delta T$ use the mass of the **water/solution**, then divide by moles and add the minus sign for an exothermic reaction.
- Draw **Hess cycles** with arrows the same way round, follow the alternative route, and always give ΔH a sign and units (kJ mol^{-1}).

Electrochemistry

A-Level Chemistry

Oxidation number

The **oxidation number** (also called the oxidation state) shows how many **electrons** an atom has lost or gained compared with the free element. You work it out using simple rules:

- an uncombined element has an oxidation number of 0.
- a simple ion has an oxidation number equal to its charge (so Mg^{2+} is +2).
- Group 1 is always +1, Group 2 is always +2.
- hydrogen is +1 (but −1 in metal hydrides).
- oxygen is −2 (but −1 in peroxides).
- fluorine is always −1.
- the oxidation numbers in a neutral compound add up to 0; in an ion they add up to the charge.

A Roman numeral shows the size of the oxidation number of an element, for example iron(II) means +2 and manganese(VII) in $KMnO_4$ means +7.

known	unknown	known
K	Mn	O_4
+1	+7	$4 \times (-2)$

they sum to 0: $(+1) + \text{Mn} + (-8) = 0$, so $\text{Mn} = +7$

Assigning an oxidation number: fix the known atoms (K = +1, O = −2 each), then use "they sum to zero" to find the unknown (Mn = +7)

Worked example. Find the oxidation number of sulfur in the sulfate ion, SO_4^{2-}.

Each oxygen is −2, so the four oxygens give $4 \times (-2) = -8$. In an ion the numbers add up to the charge, here −2. If sulfur is x:

$$x + (-8) = -2 \quad \Rightarrow \quad x = +6.$$

Worked example. Find the oxidation number of nitrogen in the nitrate ion, NO_3^-.

$$x + 3 \times (-2) = -1 \quad \Rightarrow \quad x - 6 = -1 \quad \Rightarrow \quad x = +5.$$

Redox in terms of electrons



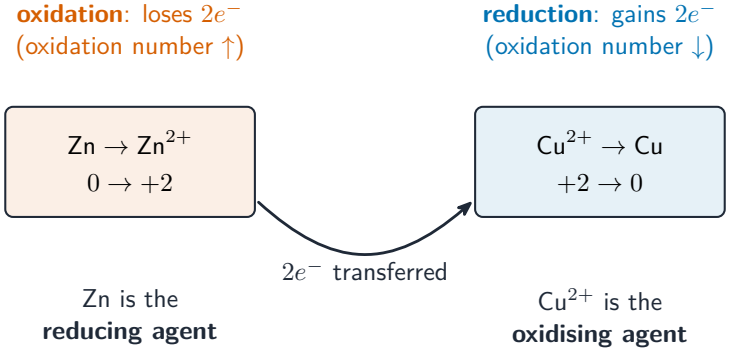
A battery uses redox reactions to push electrons round a circuit.

A **redox** reaction is one where electrons move from one species to another.

- **oxidation** is the **loss** of electrons. The oxidation number goes **up**.
- **reduction** is the **gain** of electrons. The oxidation number goes **down**.

A useful memory aid is **OIL RIG**: Oxidation Is Loss, Reduction Is Gain.

Oxidation and reduction always happen together, because the electrons lost by one species are gained by another. This **electron transfer** is why we call it a redox reaction.



Redox is electron transfer: the reducing agent loses electrons (oxidised, number up); the oxidising agent gains them (reduced, number down)

Using oxidation numbers to balance equations

Changes in oxidation number help you balance a redox equation:

1. find the element whose oxidation number **rises** (it is oxidised) and the one whose number **falls** (it is reduced).
2. the total rise must equal the total fall, because every electron lost is gained somewhere.
3. choose the ratio of the two species so the rise and fall match, then balance the rest of the equation.

$\mathrm{Fe^{2+}} \rightarrow \mathrm{Fe^{3+}}$ $(+2 \rightarrow +3)$ — oxidised: rises 1 $(\times 5)$

$\mathrm{MnO_4^-} \rightarrow \mathrm{Mn^{2+}}$ $(+7 \rightarrow +2)$ — reduced: falls 5 $(\times 1)$

5 electrons lost = 5 gained $\Rightarrow$ ratio **5 $\mathrm{Fe^{2+}}$: 1 $\mathrm{MnO_4^-}$**

Balancing a redox equation: the total rise in oxidation number must equal the total fall, which fixes the ratio

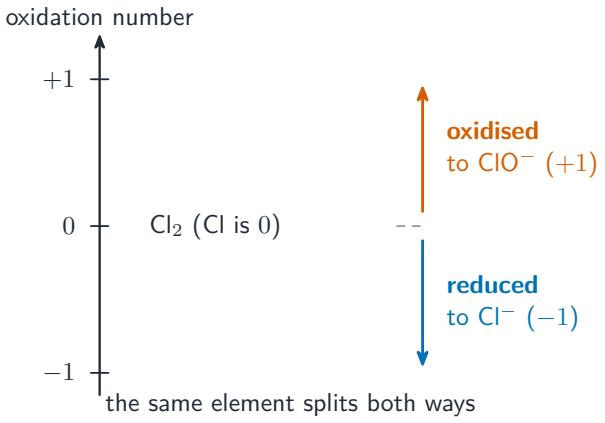
> **Worked example.** Balance the reaction of manganate(VII) with iron(II) in acid: $\mathrm{MnO_4^- + Fe^{2+} + H^+ \rightarrow Mn^{2+} + Fe^{3+} + H_2O}$.

Manganese falls from +7 to +2 (a fall of 5); iron rises from +2 to +3 (a rise of 1). To make the total fall equal the total rise, take 5 iron ions for every 1 manganate ion. Then balance oxygen with water ($4\,\mathrm{H_2O}$) and hydrogen with $\mathrm{H^+}$ ($8\,\mathrm{H^+}$):

$$\mathrm{MnO_4^- + 5Fe^{2+} + 8H^+ \rightarrow Mn^{2+} + 5Fe^{3+} + 4H_2O}.$$

Disproportionation

Disproportionation is a special redox reaction in which the **same** element is both oxidised and reduced at the same time. For example, when chlorine reacts with cold water, some chlorine atoms are reduced (to $\mathrm{Cl^-}$) and others are oxidised (to $\mathrm{ClO^-}$).



Disproportionation: chlorine (0) is both oxidised to ClO^- (+1) and reduced to Cl^- (−1) at once

Oxidising and reducing agents



Rusting is the oxidation of iron by oxygen.

- an **oxidising agent** takes electrons away from another species. In doing so, it is itself reduced.
- a **reducing agent** gives electrons to another species. In doing so, it is itself oxidised.

So in any redox reaction, the oxidising agent causes the oxidation of the other species, while the reducing agent causes the reduction.

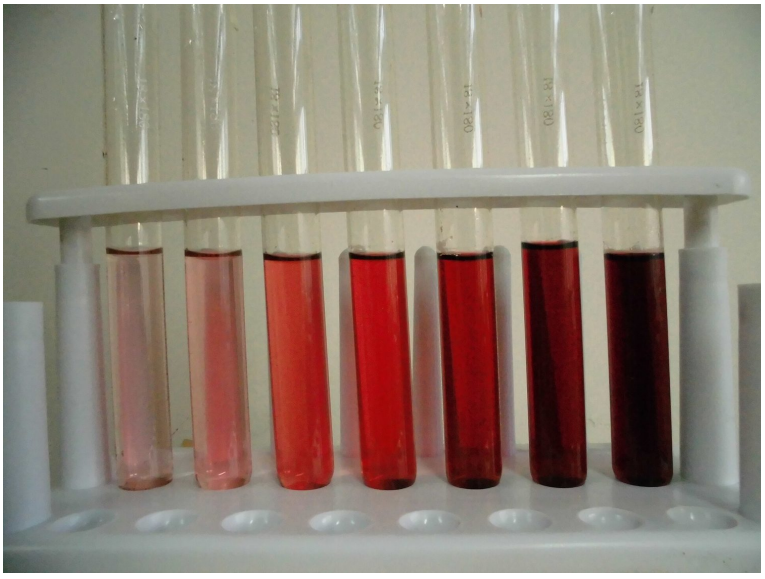
Exam tips

- Apply the oxidation-number rules in order: O is -2 **except** in peroxides (-1); H is $+1$ **except** in metal hydrides (-1).
- In a redox equation the **total increase equals the total decrease** in oxidation number – use this to fix the ratio, then balance O with H_2O and H with H^+.
- An **oxidising agent is itself reduced**; name both what is oxidised/reduced *and* the agent for full marks.
- **Disproportionation** is the same element both oxidised and reduced – show both oxidation-number changes.

Equilibria

A-Level Chemistry

Reversible reactions and dynamic equilibrium



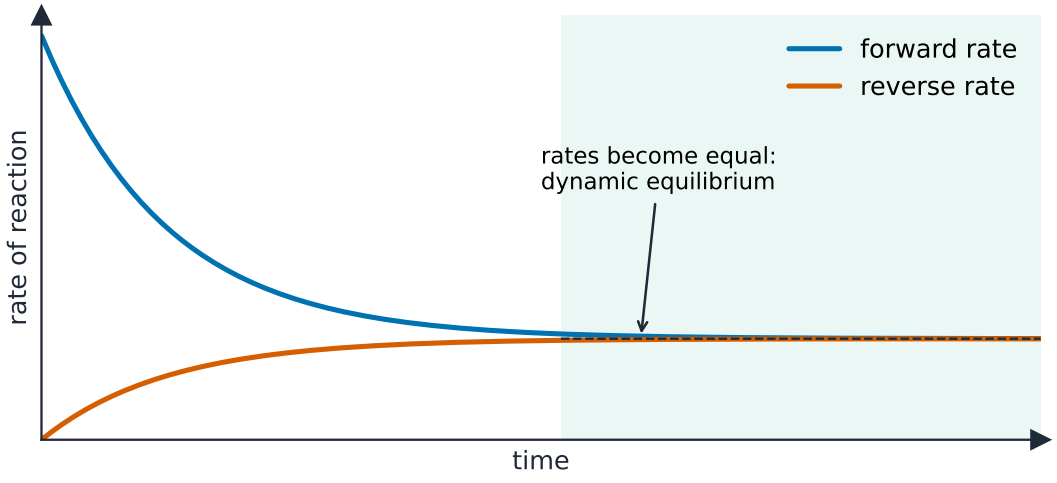
Cobalt chloride changes colour as its equilibrium shifts.

A **reversible reaction** can go both ways. The **forward reaction** makes products; the **reverse reaction** turns products back into reactants. We show this with the sign $\rightleftharpoons$.

In a **closed system** (nothing enters or leaves), the reaction reaches a **dynamic equilibrium** when:

- the rate of the forward reaction equals the rate of the reverse reaction.
- the concentrations of reactants and products stay constant.

It is called *dynamic* because both reactions are still happening – they just cancel out. A closed system is needed, or products would escape and equilibrium could never be reached.



at equilibrium: $\text{rate}_f = \text{rate}_r$ · $K_c = \frac{[\text{products}]}{[\text{reactants}]}$ (stoich. powers)

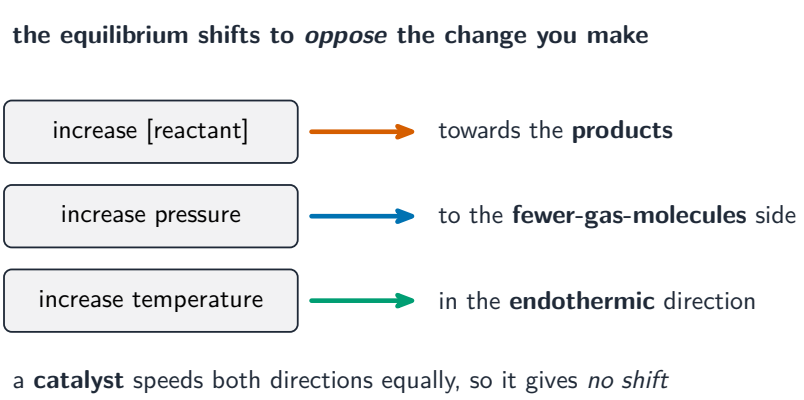
At dynamic equilibrium the forward and reverse rates have become equal, so the concentrations stay constant

Le Chatelier's principle

Le Chatelier's principle says: if you change a system at equilibrium, the position of **equilibrium** moves to oppose (reduce) that change.

Change you make	Which way the equilibrium moves
increase concentration of a reactant	towards the products
increase pressure	towards the side with fewer gas molecules
increase temperature	towards the **endothermic** direction
add a **catalyst**	no shift (it speeds up both ways equally)

A catalyst lets equilibrium be reached faster, but it does **not** change the position of equilibrium.



Le Chatelier's principle: the equilibrium always shifts to oppose the change you make

Equilibrium constants

For a reaction at equilibrium, the **equilibrium constant** links the amounts of products and reactants. For the reaction $aA + bB \rightleftharpoons cC + dD$:

$$K_c = \frac{[C]^c\,[D]^d}{[A]^a\,[B]^b}$$

where the square brackets mean concentration in mol dm^{-3}.

> **Worked example.** At equilibrium the mixture $H_2 + I_2 \rightleftharpoons 2HI$ has $[H_2] = 0.20$, $[I_2] = 0.20$ and $[HI] = 1.6$ mol dm^{-3}. Find K_c.

$$K_c = \frac{[HI]^2}{[H_2][I_2]} = \frac{1.6^2}{0.20 \times 0.20} = 64$$

(no units here, because the concentration powers cancel top and bottom).

Worked example (ICE table). 1.00 mol each of H_2 and I_2 are sealed in a 1.00 dm^3 flask; at equilibrium 0.20 mol of H_2 remains. Set out an **ICE table** (Initial, Change, Equilibrium):

	H_2	I_2	$2HI$
Initial	1.00	1.00	0
Change	−0.80	−0.80	+1.60
Equilibrium	0.20	0.20	1.60

0.80 mol of H_2 reacted, so $2 \times 0.80 = 1.60$ mol HI formed. In the 1.00 dm^3 flask the concentrations equal the moles, so $K_c = \dfrac{1.60^2}{0.20 \times 0.20} = 64$.

For reactions of gases, we use **partial pressure** instead of concentration. The partial pressure of a gas is the share of the total pressure that it provides. It is found from the **mole fraction** (the fraction of all the moles that are that gas):

$$\text{partial pressure} = \text{mole fraction} \times \text{total pressure}$$

The constant written with partial pressures is K_p.

> **Worked example.** A gas mixture holds 2.0 mol of N_2 and 6.0 mol of H_2 at a total pressure of 200 kPa. Find the partial pressure of each gas.

There are 8.0 mol in total, so

$$p(N_2) = \frac{2.0}{8.0} \times 200 = 50 \text{ kPa}, \qquad p(H_2) = \frac{6.0}{8.0} \times 200 = 150 \text{ kPa}.$$

Worked example (a numeric K_p). For $N_2 + 3H_2 \rightleftharpoons 2NH_3$ the equilibrium partial pressures are $p(N_2) = 20$, $p(H_2) = 40$ and $p(NH_3) = 10$ kPa. Then

$$K_p = \frac{p(NH_3)^2}{p(N_2)\,p(H_2)^3} = \frac{10^2}{20 \times 40^3} = 7.8 \times 10^{-5} \text{ kPa}^{-2},$$

the units coming from $\dfrac{\text{kPa}^2}{\text{kPa} \times \text{kPa}^3} = \text{kPa}^{-2}$.

Only **temperature** changes the value of K_c or K_p. Changing concentration or pressure shifts the position of equilibrium but leaves the constant unchanged; adding a catalyst changes **neither** the constant nor the position – it only makes equilibrium arrive faster.

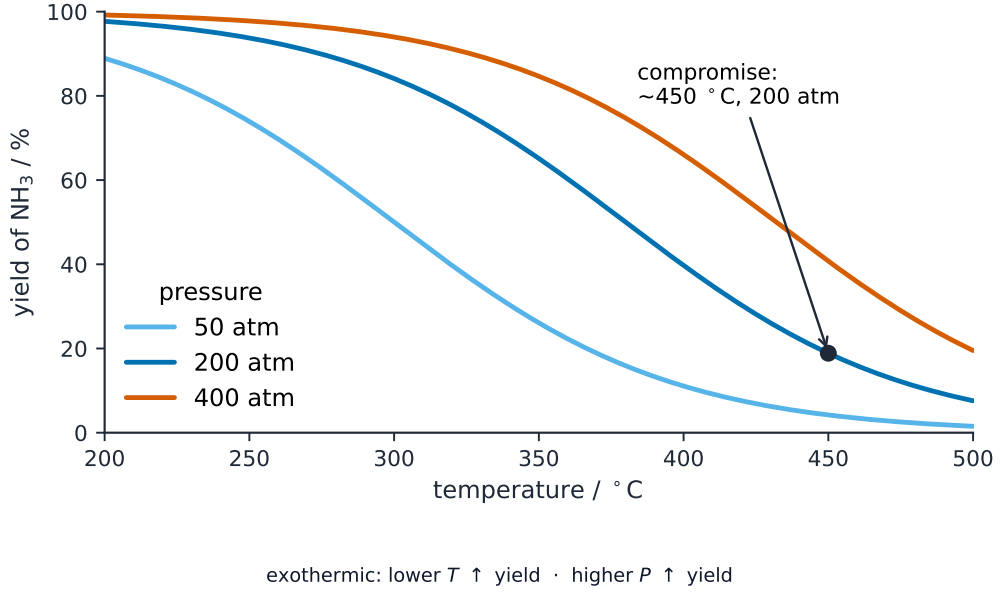
The Haber and Contact processes



Ammonia for fertiliser is made by the Haber process in large industrial plants like this one.

These two industrial processes are chosen by balancing yield, rate and cost using Le Chatelier's principle.

- the **Haber process** makes ammonia: $N_2 + 3H_2 \rightleftharpoons 2NH_3$ (exothermic). It uses about 450 °C, 200 atm and an iron catalyst. A low temperature would give more ammonia but too slowly, so a moderate temperature is a compromise.
- the **Contact process** makes sulfur trioxide: $2SO_2 + O_2 \rightleftharpoons 2SO_3$ (exothermic). It uses about 450 °C, near 1–2 atm and a vanadium(V) oxide catalyst.

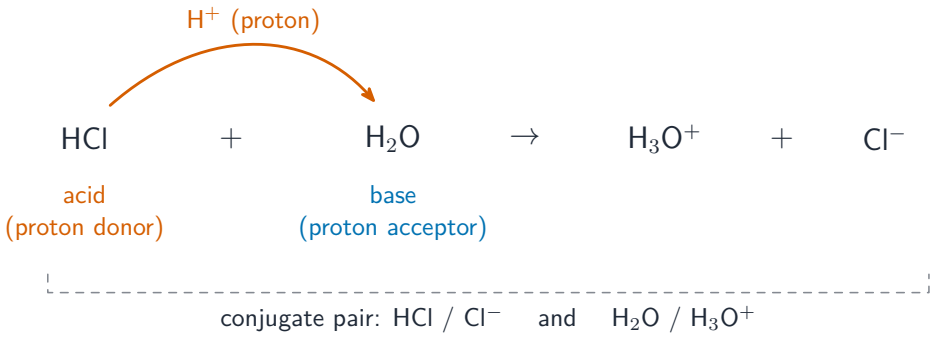


The Haber equilibrium gives more ammonia at lower temperature and higher pressure; about 450 °C and 200 atm is the working compromise

Acids and bases: the Brønsted–Lowry theory

A **proton** is simply an H^+ ion. The Brønsted–Lowry theory defines acids and bases by what they do with protons:

- an **acid** is a **proton donor** (it gives away H^+).
- a **base** is a **proton acceptor** (it takes H^+). A base that dissolves in water is called an alkali.



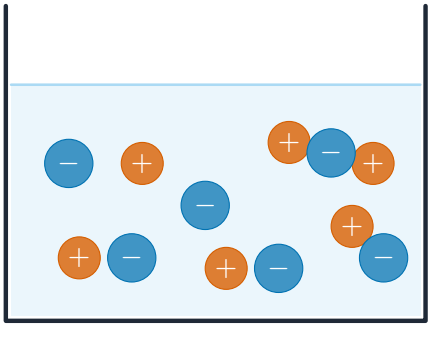
Brønsted–Lowry: the acid donates a proton (H^+) to the base, making two conjugate acid–base pairs

Common acids you must know: hydrochloric acid (HCl), sulfuric acid (H_2SO_4), nitric acid (HNO_3) and ethanoic acid (CH_3COOH). Common alkalis: sodium hydroxide (NaOH), potassium hydroxide (KOH) and ammonia (NH_3).

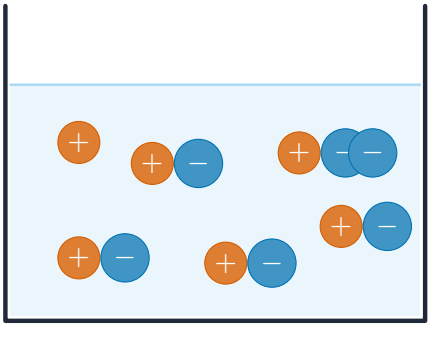
Strong and weak acids and bases

This is about how fully an acid or base splits up in water – not how concentrated it is.

- a **strong acid** or **strong base** is fully **dissociated** in water (almost every molecule splits into ions).
- a **weak acid** or **weak base** is only partly dissociated (most molecules stay whole).



strong acid
fully dissociated: many ions



weak acid
mostly molecules: few ions

Same concentration, different ionisation: a strong acid is fully dissociated into ions; a weak acid stays mostly as whole molecules

The **pH** scale measures how acidic a solution is: pure water is pH 7, acids are below

7, and alkalis are above 7. A strong acid has a lower pH than a weak acid of the same concentration.

You can tell strong and weak acids apart by:

- **reaction with a reactive metal**: a strong acid fizzes faster.
- **pH**: measured with a pH meter or **universal indicator** .
- **electrical conductivity**: a strong acid conducts better, because it has more ions.

Neutralisation, salts and titration curves

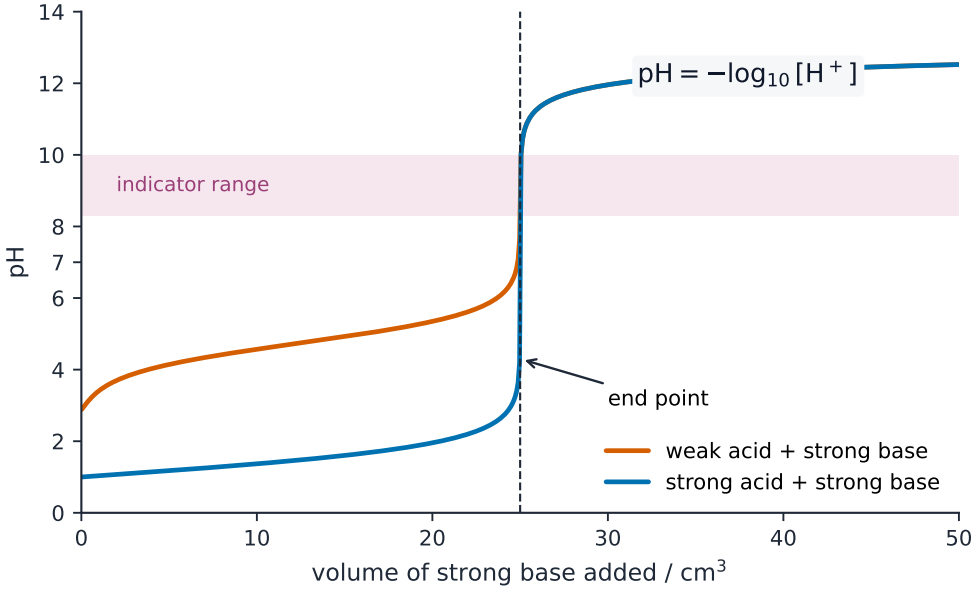
Neutralisation happens when the hydrogen ions from an acid react with the hydroxide ions from an alkali:

$$H^+(aq) + OH^-(aq) \rightarrow H_2O(l)$$

A **salt** is also formed, from the rest of the acid and base.

In a **titration** you add one solution to another and follow the pH. The **titration curve** has a steep, almost vertical jump near the end point. The exact shape depends on whether each reactant is strong or weak.

To pick an **indicator** , choose one whose colour change falls inside that steep jump. For a strong acid with a strong base most indicators work; for a weak acid with a strong base you need one that changes in the higher pH range.



Titration curves each have a steep pH jump at the end point. The weak-acid jump sits higher, so the indicator must change colour in that range

Exam tips

- Define **dynamic equilibrium** with all three points: forward and reverse rates equal, concentrations constant, closed system.
- Answer **Le Chatelier** questions by stating the change, the direction of shift, and the reason; a **catalyst does not shift** the position (it speeds both rates).
- Write K_c as **products over reactants**, each raised to its balancing number; include state symbols to decide what appears.
- Justify Haber/Contact conditions as a **compromise** between yield, rate and cost – do not just quote them.
- Brønsted: acid = **proton donor**, base = proton acceptor; conjugate pairs differ by one H^+.

Reaction kinetics

A-Level Chemistry

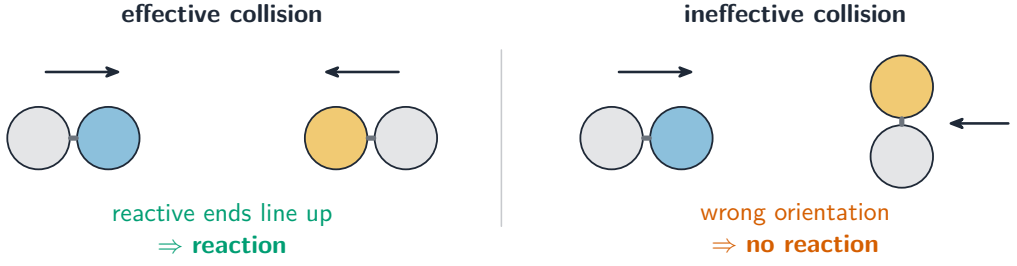
Rate of reaction

The **rate of reaction** is how fast reactants turn into products. We measure it as the change in concentration (or amount) in each unit of time.

To react, particles must **collide** . The **collision frequency** is how often the particles hit each other. But not every collision leads to a reaction:

- an **effective collision** has enough energy *and* the correct direction, so a reaction happens.
- a **non-effective collision** does not have enough energy, or the particles hit at the wrong angle, so nothing happens.

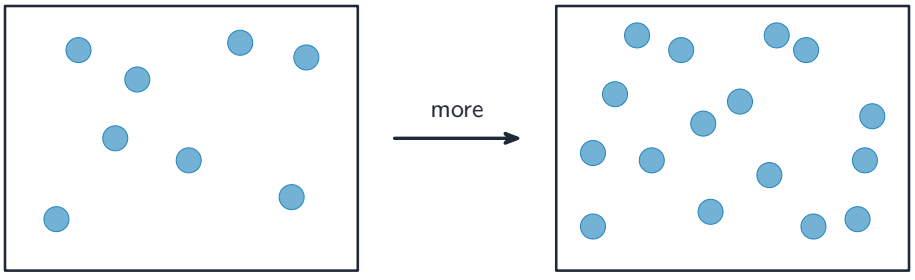
So the rate depends on the **frequency of effective collisions** – how many useful collisions happen each second.



A collision only reacts with the right orientation and enough energy (coloured ends = the reactive part)

Concentration and pressure

If you increase the concentration of a solution (or the pressure of a gas), the particles are packed closer together. They collide more often, so there are more effective collisions each second, and the rate goes up.



More particles in the same volume collide more often, so the rate rises

You can calculate a rate from experimental data – for example, the volume of gas made divided by the time taken.

> **Worked example.** A reaction gives off carbon dioxide. In the first 30 s, 48 cm^3 of gas is collected. Find the average rate of reaction over this time.

$$\text{average rate} = \frac{\text{volume of gas}}{\text{time}} = \frac{48}{30} = 1.6 \text{ cm}^3\,\text{s}^{-1}.$$

The rate is fastest at the start (the graph is steepest there) because the reactants are most concentrated, then it slows as they are used up.

Temperature and activation energy

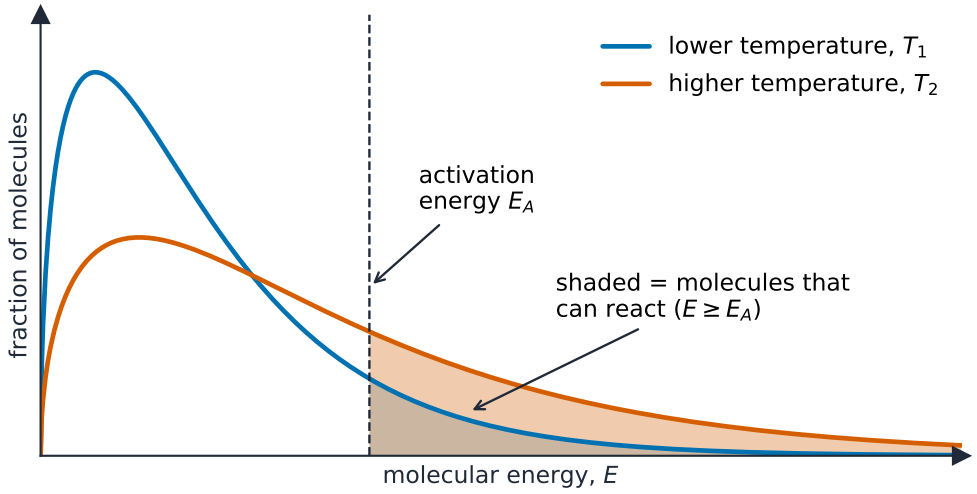
The **activation energy** (E_A) is the minimum energy a collision needs in order to be effective.

The **Boltzmann distribution** is a graph showing how the energies of the molecules are spread out at one temperature. The curve starts at the origin, rises to a peak, then falls away in a long tail. The total area under the curve is the total number of molecules. Only the molecules to the right of E_A have enough energy to react.

When you raise the temperature:

- the curve flattens and spreads to the right, so a **much larger fraction** of molecules now have energy greater than E_A.
- the molecules also move faster and collide more often.

The first effect is the bigger one. This is why a small rise in temperature gives a large rise in rate.



higher $T \Rightarrow$ more molecules with $E \geq E_A$

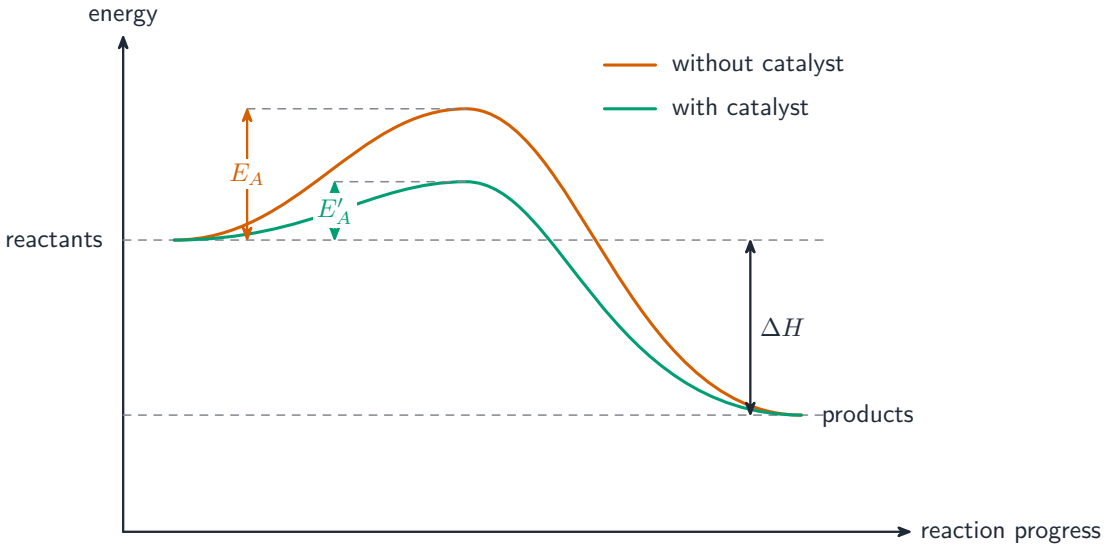
At higher temperature the curve spreads to the right, so a larger fraction of molecules can react

Catalysts

A **catalyst** speeds up a reaction but is not used up itself. **Catalysis** is the name for this action.

A catalyst works by giving the reaction a different **reaction mechanism** – a new route with a **lower** activation energy. On the Boltzmann distribution, lowering E_A moves the line to the left, so more molecules now have enough energy. This means more effective collisions each second, and a faster rate.

On a **reaction pathway diagram**, the catalysed route has a lower energy "hill". The enthalpy change of the reaction, ΔH, is **not** changed by the catalyst.



A catalyst gives a route with lower activation energy; the enthalpy change is unchanged

There are two types:

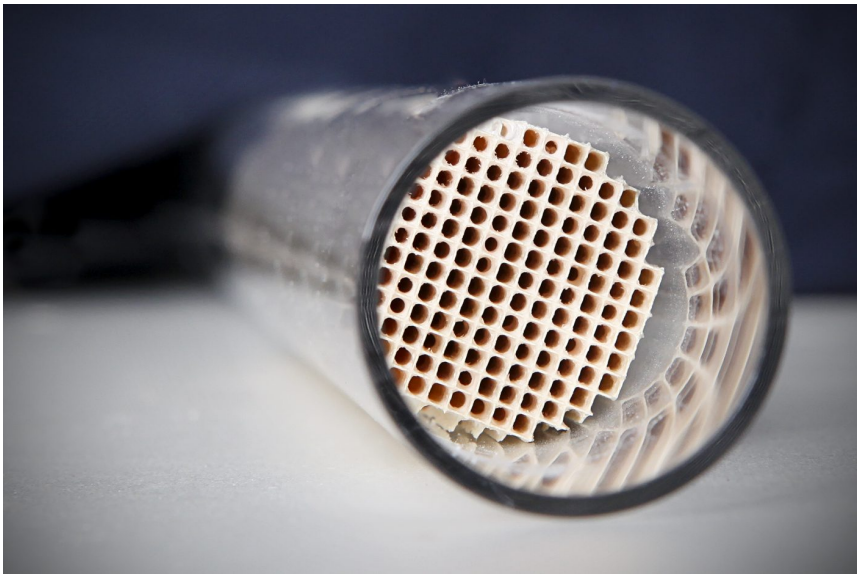
- a **homogeneous catalyst** is in the **same** physical state as the reactants – for example, an acid catalyst dissolved in a solution of liquids.
- a **heterogeneous catalyst** is in a **different** state from the reactants – for example, solid iron speeding up the reaction of gases in the Haber process.



homogeneous (same state, mixed in)

heterogeneous (different state, a surface)

A homogeneous catalyst is mixed in with the reactants (same state); a heterogeneous catalyst is a separate surface (different state), where reactants meet and react



A car's catalytic converter is a heterogeneous catalyst; its honeycomb gives a huge surface area

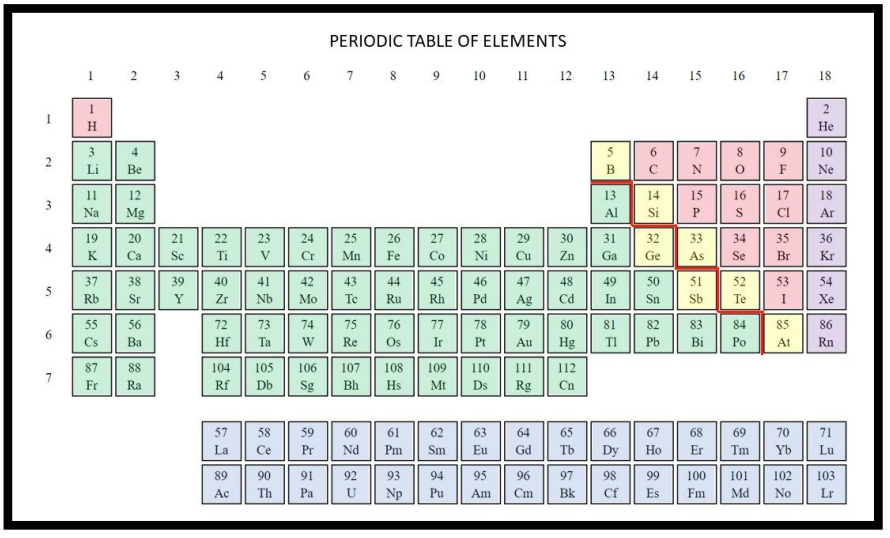
Exam tips

- Explain rate changes with **collision theory** – more frequent and/or more energetic **effective collisions**.
- On a **Boltzmann distribution** mark E_A, shade to its right, and show the curve **flatten and shift right** at higher temperature; it starts at the origin and never touches the axis.
- A **catalyst** gives an alternative route with **lower** E_A and does **not** change ΔH.
- Say why a small temperature rise gives a large rate rise: a **much greater proportion** of molecules now exceed E_A (the main effect).

The Periodic Table: chemical periodicity

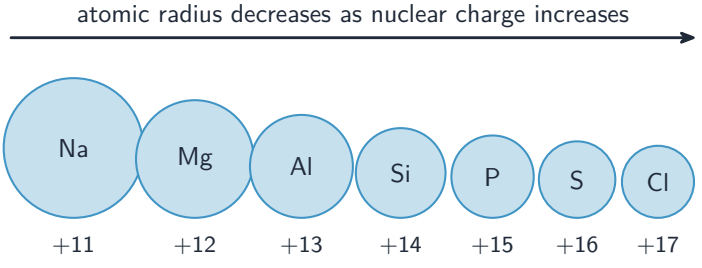
A-Level Chemistry

Physical properties across Period 3



Properties repeat in a regular pattern across each period of the table.

Periodicity means that properties repeat in a regular pattern as you go across each **period** of the Periodic Table. Period 3 (Na to Ar) is the standard example.



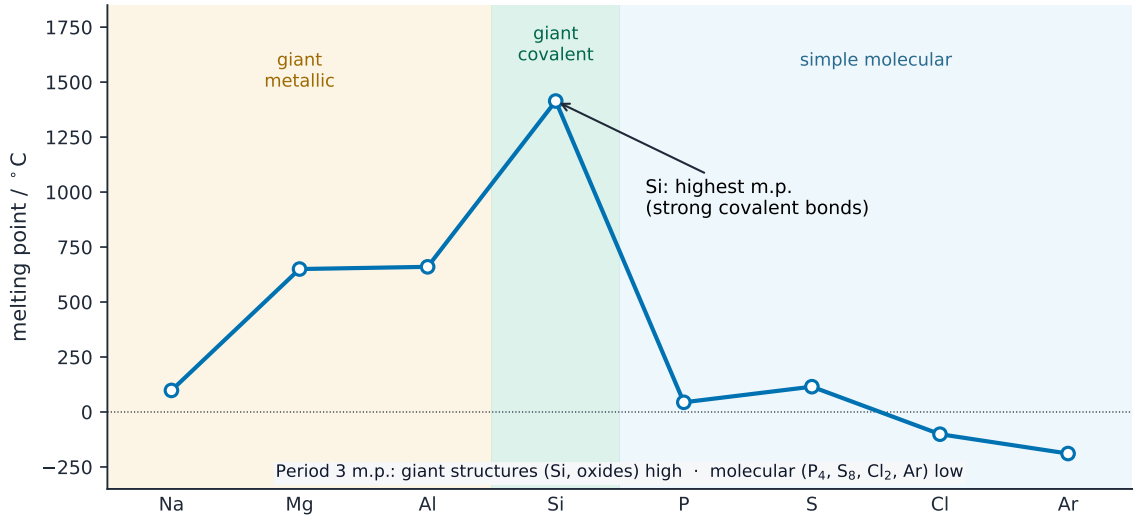
Atomic radius decreases across Period 3: the rising nuclear charge pulls the same outer shell inwards

Property	Trend across Period 3
atomic radius	gets smaller (more nuclear charge pulls the same shell in)
ionic radius	positive ions are small; from P^{3-} onwards the negative ions are larger
melting point	rises to a peak at silicon, then falls sharply
electrical conductivity	high for Na, Mg, Al; almost zero from Si onwards

The melting point and conductivity follow from the structure and bonding:

- **Na, Mg, Al** are **giant metallic** . Melting points rise (Na → Al) because each atom gives more delocalised electrons and the ions get smaller, so the bonding is stronger. They conduct well.
- **Si** is **giant molecular** (giant covalent). It has the highest melting point, because strong covalent bonds must be broken. It barely conducts.
- **P, S, Cl, Ar** are **simple molecular** (or single atoms). Their melting points are low, because the only forces that break are the weak **van der Waals' forces** **between** the molecules - the strong covalent bonds **inside** each molecule are never broken at all. They do not conduct.

Name **van der Waals' forces** when you explain that drop: it is the marking point, and "the covalent bonds are weak" is the error that loses it. Melting S_8 pulls whole molecules apart from each other; it does not touch a single S–S bond. Their sizes still follow the molecule: S_8 melts higher than P_4 because it is bigger, with more electrons and so stronger van der Waals' forces, while Ar (a single atom) is lowest of all.



Melting point across Period 3 peaks at silicon (giant covalent); it is high for the metals and low for the simple molecular elements

Chemical properties across Period 3

■ Reactions with oxygen, chlorine and water

With **oxygen**:

$$4Na + O_2 \rightarrow 2Na_2O \qquad 2Mg + O_2 \rightarrow 2MgO \qquad 4Al + 3O_2 \rightarrow 2Al_2O_3$$

$$P_4 + 5O_2 \rightarrow P_4O_{10} \qquad S + O_2 \rightarrow SO_2$$

With **chlorine**:

$$2Na + Cl_2 \rightarrow 2NaCl \qquad Mg + Cl_2 \rightarrow MgCl_2 \qquad 2Al + 3Cl_2 \rightarrow 2AlCl_3$$

$$Si + 2Cl_2 \rightarrow SiCl_4 \qquad 2P + 5Cl_2 \rightarrow 2PCl_5$$

With **water** (only Na and Mg react):

$$2Na + 2H_2O \rightarrow 2NaOH + H_2 \qquad Mg + 2H_2O \rightarrow Mg(OH)_2 + H_2$$

Sodium reacts fast; magnesium reacts only very slowly with cold water.

■ Oxidation number of the oxides and chlorides

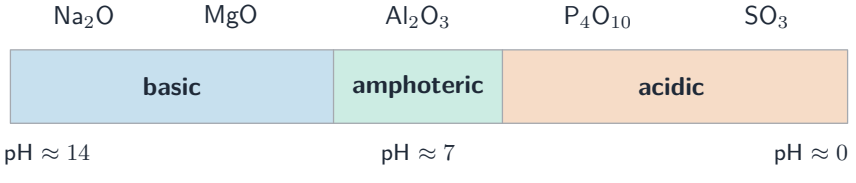
The **oxidation number** of the Period 3 element in its oxide or chloride rises across the period, because it equals the number of outer-shell (**valence shell**) **electrons** the atom uses in bonding:

Oxides	**Na_2O**	**MgO**	**Al_2O_3**	**P_4O_{10}**	**SO_2 / SO_3**
oxidation number	+1	+2	+3	+5	+4 / +6

The chlorides NaCl, $MgCl_2$, $AlCl_3$, $SiCl_4$, PCl_5 show oxidation numbers +1 to +5 in the same way.

■ Oxides with water, and acid–base behaviour

Across the period the **oxides** change from basic to acidic:



metal oxides (left) are basic; non-metal oxides (right) are acidic

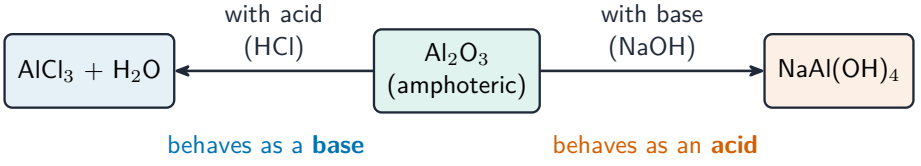
The Period 3 oxides change from basic (the metals) through amphoteric (Al_2O_3) to acidic (the non-metals)

Oxide	With water	Acid–base nature	Approximate pH
Na_2O	$Na_2O + H_2O \rightarrow 2NaOH$	basic	13–14
MgO	$MgO + H_2O \rightarrow Mg(OH)_2$	basic	9–10
Al_2O_3	insoluble	**amphoteric**	7
SiO_2	insoluble	weakly acidic	7
P_4O_{10}	$P_4O_{10} + 6H_2O \rightarrow 4H_3PO_4$	acidic	1–2
SO_3	$SO_3 + H_2O \rightarrow H_2SO_4$	strongly acidic	0–1

Metal oxides (left) are basic; non-metal oxides (right) are acidic. Al_2O_3 and its **hydroxide** $Al(OH)_3$ are amphoteric – they react with **both** acids and bases:

$$Al_2O_3 + 6HCl \rightarrow 2AlCl_3 + 3H_2O$$

$$Al_2O_3 + 2NaOH + 3H_2O \rightarrow 2NaAl(OH)_4$$

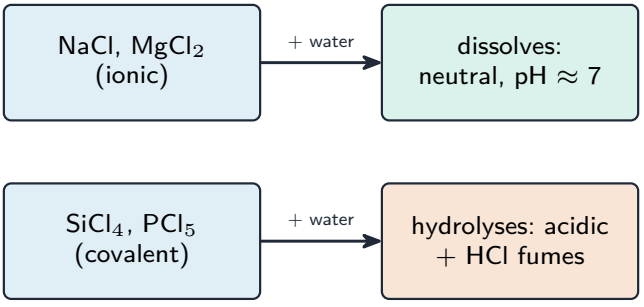


Aluminium oxide is amphoteric – it reacts with acids (behaving as a base) and with bases (behaving as an acid)

Chlorides with water

- NaCl and $MgCl_2$ are ionic. They simply dissolve, giving a near-neutral solution.
- $SiCl_4$ and PCl_5 are covalent. They undergo **hydrolysis** (react with water) to make acidic solutions and fumes of HCl:

$$SiCl_4 + 2H_2O \rightarrow SiO_2 + 4HCl \qquad PCl_5 + 4H_2O \rightarrow H_3PO_4 + 5HCl$$



The ionic chlorides just dissolve (neutral); the covalent chlorides react with water (hydrolyse) to give an acidic solution and HCl fumes

■ Explaining the trends

These trends follow from the change in bonding and **electronegativity** . On the left, the elements are metals with low electronegativity, so their oxides and chlorides are ionic and basic (or neutral). On the right, the elements are non-metals with high electronegativity, so their oxides and chlorides are covalent and acidic. You can use a chloride's or oxide's properties (melting point, conductivity, effect on water) to suggest whether its bonding is ionic or covalent.

> **Worked example.** Predict the pH when Na_2O, Al_2O_3 and SO_3 are each added to water. Acid-base character follows the **metal to non-metal** change across Period 3. Na_2O is ionic and basic: it dissolves to give NaOH, so the pH is about **13**. SO_3 is covalent and acidic: it reacts to give H_2SO_4, so the pH is about **1**. Al_2O_3 sits at the changeover - it is **amphoteric** and essentially **insoluble** in water, so the pH stays about **7**. The trap is that last one: amphoteric does **not** mean neutral, it means the oxide reacts with acids **and** with alkalis - it simply does not react with water.

Periodicity of other elements

The same idea works for any group. If you know the pattern down a group and across a period, you can:

- **predict** the properties of an element from its position (for example, a Group 1 element will be a reactive metal forming a +1 ion).
- **deduce** the likely position and identity of an unknown element from its physical and chemical properties.

Exam tips

- Always pair the **trend with its reason**: atomic radius falls across Period 3 (rising nuclear charge, similar shielding).
- Explain the **melting-point pattern** by structure: giant metallic (Na→Al), giant covalent (Si, highest), then simple molecular (P_4, S_8, Cl_2) and monatomic (Ar).
- Oxides go **basic → amphoteric (Al_2O_3) → acidic** across the period; link to ionic-to-covalent bonding.
- For reactions of oxides/chlorides with water, give the **products and the resulting pH**.

Group 2

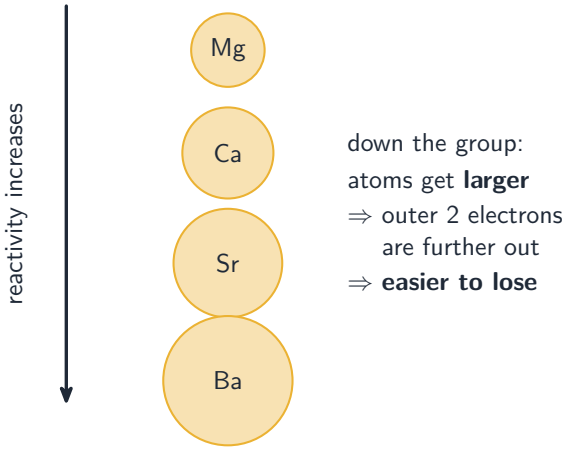
A-Level Chemistry

The Group 2 metals



Limestone is calcium carbonate – a compound of the Group 2 metal calcium.

Group 2 holds the metals magnesium, calcium, strontium and barium. They all have two outer electrons, which they lose to form 2+ ions. Going **down** the group, the atoms get larger and the outer electrons are easier to lose, so the metals get **more reactive** – their **reactivity** increases down the group.



Reactivity increases down Group 2: larger atoms hold their two outer electrons less tightly, so they are lost more easily

Reactions of the elements

With **oxygen**, they burn to form an oxide:

$$2Mg + O_2 \rightarrow 2MgO$$



Magnesium burns in air with a brilliant white flame, forming white magnesium oxide

With **water**, they form a hydroxide and hydrogen. The reaction gets faster down the group:

$$Ca + 2H_2O \rightarrow Ca(OH)_2 + H_2$$

Magnesium is slow with cold water but reacts fast with steam, giving MgO and H_2.

With **dilute hydrochloric or sulfuric acid**, they form a salt and hydrogen:

$$Mg + 2HCl \rightarrow MgCl_2 + H_2 \qquad Mg + H_2SO_4 \rightarrow MgSO_4 + H_2$$

With sulfuric acid the reaction slows down the group, because the **sulfates** formed (such as $BaSO_4$) are insoluble and coat the metal.

Reactions of the compounds

The **oxides** and **hydroxides** are basic. They react with water and with dilute acids:

$$MgO + H_2O \rightarrow Mg(OH)_2 \qquad Mg(OH)_2 + 2HCl \rightarrow MgCl_2 + 2H_2O$$

The **carbonates** react with dilute acids to give a salt, water and carbon dioxide:

$$CaCO_3 + 2HCl \rightarrow CaCl_2 + H_2O + CO_2$$

Thermal decomposition

Thermal decomposition means breaking a compound apart by heating it.

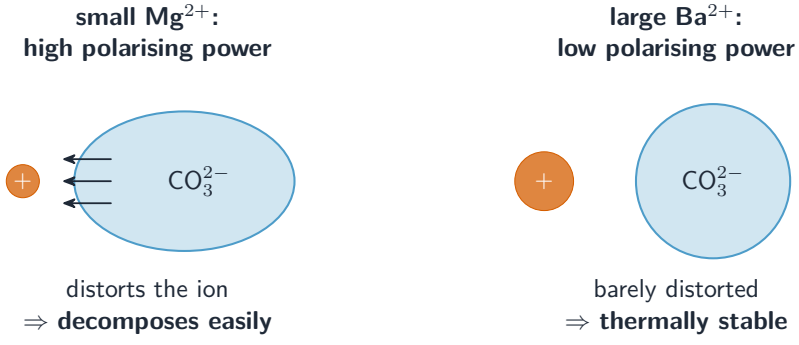
- the carbonates break into the oxide and carbon dioxide:

$$CaCO_3 \rightarrow CaO + CO_2$$

- the **nitrates** break into the oxide, brown nitrogen dioxide gas and oxygen:

$$2Ca(NO_3)_2 \rightarrow 2CaO + 4NO_2 + O_2$$

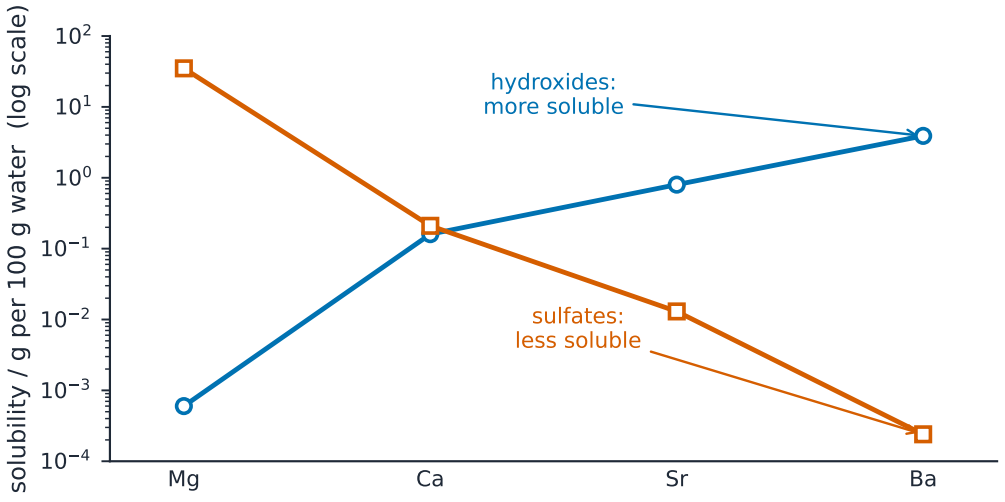
The **thermal stability** of both the carbonates and the nitrates **increases** down the group. A larger metal ion pulls less on the carbonate or nitrate ion, so the compound is harder to break apart – it needs a higher temperature. So magnesium carbonate decomposes most easily, and barium carbonate is the hardest.



Thermal stability rises down the group: a small cation polarises (distorts) the carbonate ion more, weakening it so it decomposes more easily

Trends in solubility

Compound	Trend in solubility down the group
hydroxides	**increase** ($Mg(OH)_2$ almost insoluble; $Ba(OH)_2$ soluble)
sulfates	**decrease** ($MgSO_4$ soluble; $BaSO_4$ insoluble)



Group 2: OH^- sol. ↑ down · SO_4^{2-} sol. ↓ down

Down Group 2 the hydroxides become more soluble while the sulfates become less soluble (note the log scale)

From these trends you can predict the properties of the next element down, radium, and of its compounds.

> **Worked example.** $MgCO_3$ decomposes at about 540 °C and $BaCO_3$ at about 1360 °C. Explain the trend and predict where $CaCO_3$ sits. Going **down** Group 2 the cation gets **larger**, so its charge is spread over a bigger surface and its **polarising power** falls. A less polarising cation distorts the carbonate ion less, so the C-O bond is weakened less and more heat is needed to break it. Thermal stability therefore **increases down the group**, and $CaCO_3$, lying between Mg and Ba, decomposes at an intermediate temperature of about 900 °C. Argue the whole chain - **cation size, then polarising power, then distortion of the anion**; "it is more reactive" explains nothing here.

Exam tips

- Reactivity **increases down** Group 2: ionisation energy falls as atoms get larger, so electrons are lost more easily.
- Reactions with water become **more vigorous** down the group; Mg reacts slowly with cold water but fast with steam (→ MgO).
- Hydroxides get **more soluble** down the group; sulfates get **less soluble** ($BaSO_4$ insoluble – the basis of the sulfate test).
- Write equations with the correct +2 ion and state the observations (fizzing, dissolving).

Group 17

A-Level Chemistry

Physical properties of the halogens

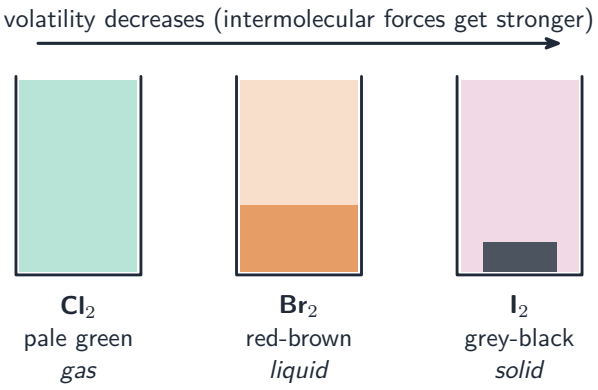


Bromine is a Group 17 halogen – a dark liquid that gives off an orange vapour.

The **halogens** are the **Group** 17 elements. They exist as diatomic molecules (Cl_2, Br_2, I_2). Going down the group:

Element	Colour and state at room temperature
chlorine	pale green gas
bromine	red-brown liquid
iodine	grey-black solid (purple vapour)

The **volatility** (how easily a substance turns to vapour) **decreases** down the group: chlorine is a gas, but iodine is a solid. This is because the molecules get larger and have more electrons, so the **instantaneous dipole** and **induced dipole** forces between them get stronger. Stronger forces are harder to break, so the boiling point rises and volatility falls.



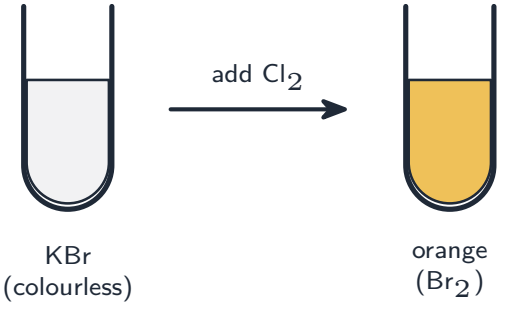
Down Group 17 the halogens change from a pale green gas to a red-brown liquid to a grey-black solid as volatility falls

The **bond energy** (bond strength) of the X–X molecules generally falls from Cl_2 to I_2, because the shared electrons are further from the nuclei in the larger atoms.

Chemical properties of the halogens and hydrogen halides

Halogens as oxidising agents

Each halogen reacts by gaining one electron to form a 1− ion, so it acts as an **oxidising agent** . This power **decreases** down the group, because the larger atoms attract an extra electron less strongly. A more reactive halogen can push out a less reactive one from its salt:



Chlorine displaces bromine: the solution turns orange

$$Cl_2 + 2KBr \rightarrow 2KCl + Br_2$$

Reactions with hydrogen

Each halogen reacts with hydrogen to form a **hydrogen halide** :

$$H_2 + Cl_2 \rightarrow 2HCl$$

The reaction gets less vigorous down the group: chlorine reacts explosively in light, bromine needs heat, and iodine reacts slowly and only partly.

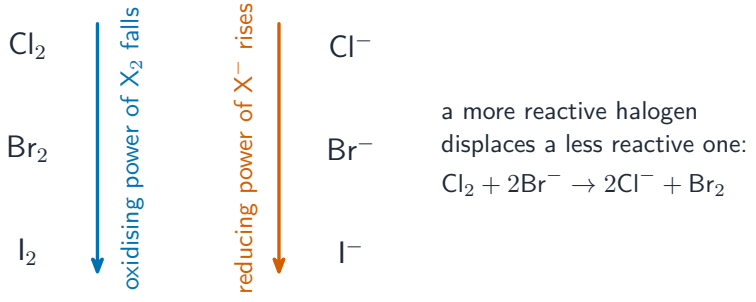
Thermal stability of the hydrogen halides

The **thermal stability** of the hydrogen halides **decreases** down the group. The H–X bond gets weaker as the halogen atom gets larger, so HI breaks apart on gentle heating while HCl is very stable.

Reactions of the halide ions

Halide ions as reducing agents

A **halide ion** (such as Cl^-) can give away an electron, acting as a **reducing agent** . This power **increases** down the group, because a larger ion holds its outer electron less tightly.

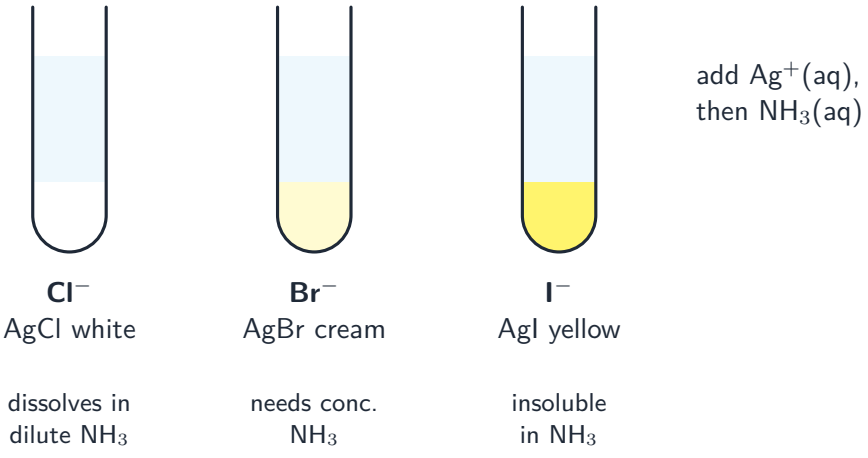


Two opposite trends: the oxidising power of the halogens falls down the group, while the reducing power of the halide ions rises

Reaction with aqueous silver ions

Add aqueous silver nitrate, then aqueous **ammonia** , to identify the halide from the colour of the silver halide **precipitate** :

Halide	Precipitate with Ag^+	Solubility in ammonia
Cl^-	white	dissolves in dilute ammonia
Br^-	cream	dissolves only in concentrated ammonia
I^-	yellow	insoluble in ammonia



Silver halide precipitates: AgCl is white, AgBr cream and AgI yellow – and their solubility in ammonia confirms which halide is present



The silver halide test: AgCl is white, AgBr cream and AgI yellow

■ Reaction with concentrated sulfuric acid

All the halides first give the hydrogen halide. The lower halides are then oxidised by the acid, because they are stronger reducing agents:

$$NaCl + H_2SO_4 \rightarrow NaHSO_4 + HCl$$

- chloride gives only HCl (no redox).
- bromide also gives some brown Br_2 and SO_2.
- iodide gives I_2 and the smelly gases H_2S and SO_2, because I^- is the strongest reducing agent.

reducing power increases

$Cl^- \rightarrow$ HCl only (no redox)

$Br^- \rightarrow$ HBr, then $Br_2 + SO_2$

$I^- \rightarrow$ HI, then $I_2 + H_2S + SO_2$

With concentrated sulfuric acid, chloride gives only HCl, but bromide and iodide (stronger reducing agents) are also oxidised to the halogen

Reactions of chlorine



Chlorine is added to pool water to kill microbes.

With sodium hydroxide

With **cold, dilute** sodium hydroxide, chlorine reacts to form chloride and chlorate(I):

$$Cl_2 + 2NaOH \rightarrow NaCl + NaClO + H_2O$$

With **hot, concentrated** sodium hydroxide, it forms chloride and chlorate(V):

$$3Cl_2 + 6NaOH \rightarrow 5NaCl + NaClO_3 + 3H_2O$$

In both, the **oxidation number** of chlorine goes both up and down (from 0), so both are **disproportionation** reactions.

■ Chlorine in water purification

A little chlorine is added to water for **water purification** . It reacts with water:

$$Cl_2 + H_2O \rightleftharpoons HOCl + HCl$$

The active species HOCl and ClO^- kill **bacteria** , making the water safe to drink.

> **Worked example.** Solid NaCl, NaBr and NaI are each warmed with concentrated H_2SO_4. Predict the products. **Reducing power increases down the group**, so how far each halide reduces the sulfuric acid differs. Cl^- is too weak to reduce it at all, so you get only steamy HCl - an acid-base reaction. Br^- reduces it a little: HBr plus brown Br_2 and SO_2. I^- is the strongest reducing agent: HI plus I_2, and it drives the sulfur all the way down to H_2S, with its bad-egg smell. **Every** halide gives the hydrogen halide first; the extra products appear only where the halide is a strong enough **reducing agent** to attack the sulfur.

Exam tips

- Halogens get **less reactive down** the group (harder to gain an electron); oxidising power decreases.
- **Displacement**: a more reactive halogen displaces a less reactive halide – state the colour change.
- Test halide ions with $AgNO_3$: white (Cl^-), cream (Br^-), yellow (I^-), then confirm with dilute/concentrated ammonia.
- Chlorine with water and with cold NaOH are **disproportionation** – show the oxidation-number changes.

Nitrogen and sulfur

A-Level Chemistry

Why nitrogen is unreactive

Nitrogen gas, N_2, makes up most of the air but reacts with very little. There are two reasons:

- the two nitrogen atoms are joined by a **triple bond** , which has a very high **bond energy** . A lot of energy is needed to break it.
- the molecule has no **polarity** – it is perfectly symmetrical, so nothing pulls other molecules towards it.



N≡N

strong triple bond: very high bond energy ⇒ hard to break

symmetrical molecule ⇒ no polarity ⇒ nothing pulls it into a reaction

Nitrogen is unreactive for two reasons: a very strong triple bond, and a symmetrical, non-polar molecule

Ammonia and the ammonium ion

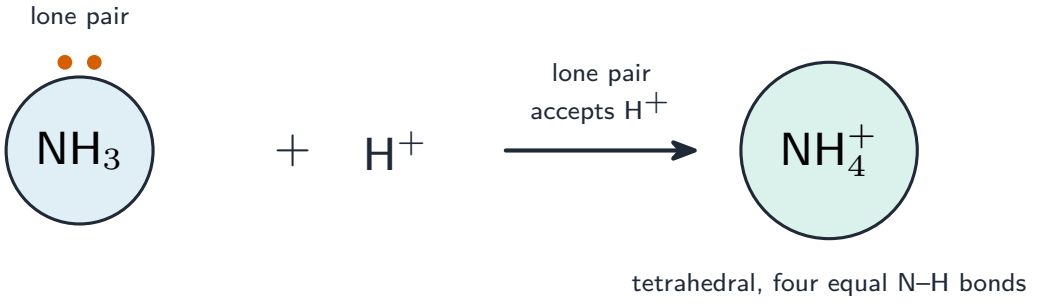


Ammonia is converted into nitrogen fertilisers in industrial plants on a huge scale.

Basicity of ammonia

The **basicity** of ammonia (its ability to act as a base) comes from the **lone pair** of electrons on the nitrogen atom. Using the Brønsted–Lowry theory, ammonia is a base because this lone pair can accept a **proton** (H^+):

$$NH_3 + H^+ \rightarrow NH_4^+$$



Ammonia acts as a base because the lone pair on its nitrogen accepts a proton, making the ammonium ion

The ammonium ion

When the lone pair forms a bond to H^+, it makes the **ammonium ion** , NH_4^+. Because both shared electrons came from the nitrogen, this new bond is a **coordinate bond** . The ion has four identical N–H bonds and a tetrahedral shape.

Displacement of ammonia from its salts

If you warm an ammonium salt with a base (such as sodium hydroxide), you push out ammonia gas. This is an acid–base **displacement** :

$$NH_4Cl + NaOH \rightarrow NaCl + NH_3 + H_2O$$

The sharp smell of ammonia, and damp red litmus turning blue, is a test for an ammonium salt.

Oxides of nitrogen and air pollution

Where they come from

Oxides of nitrogen (NO and NO_2, together called NO_x) come from two sources:

- **natural**: lightning gives enough energy for nitrogen and oxygen in the air to combine.

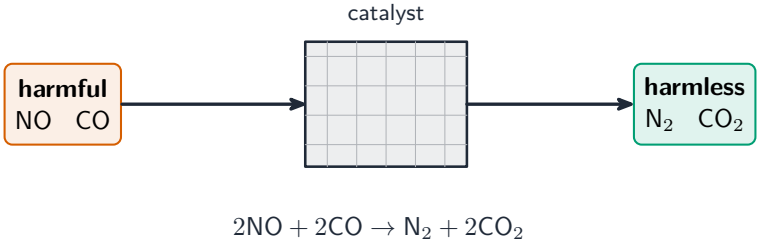
- **man-made**: the high temperature inside an **internal combustion engine** makes nitrogen and oxygen react:

$$N_2 + O_2 \rightarrow 2NO$$

Removing them from car exhaust

A **catalytic converter** cleans the **exhaust gases**. It lets the harmful gases react together to form harmless ones:

$$2NO + 2CO \rightarrow N_2 + 2CO_2$$



In a catalytic converter the harmful gases NO and CO react over the catalyst to form harmless N_2 and CO_2

Photochemical smog

In sunlight, NO and NO_2 react with unburned **hydrocarbons** to form peroxyacetyl nitrate (PAN). PAN is a harmful part of **photochemical smog**, the brown haze seen over busy cities.

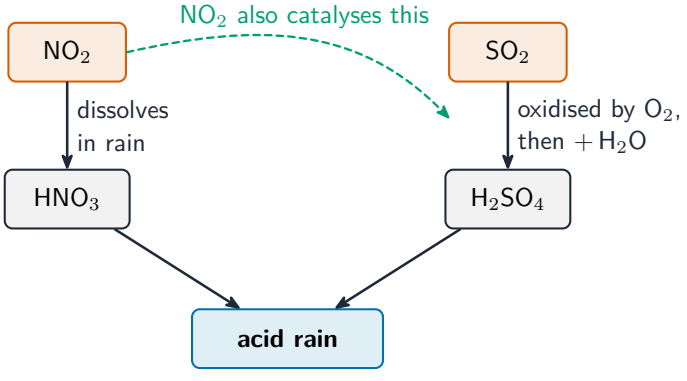


Photochemical smog over a big city. The brown layer is trapped low down, exactly where the traffic fumes are – sunlight turns those nitrogen oxides and unburned hydrocarbons into the haze you can see

Acid rain

The oxides of nitrogen also help make **acid rain** in two ways:

- directly: NO_2 dissolves in rain to form nitric acid.
- as a catalyst: NO_2 speeds up the oxidation of atmospheric **sulfur dioxide** (SO_2) into SO_3, which then forms sulfuric acid in the rain.



Nitrogen and sulfur oxides make acid rain: NO_2 forms nitric acid directly and also catalyses the oxidation of SO_2 to sulfuric acid

Worked example. A white solid is warmed with aqueous NaOH, and a gas is released that turns damp red litmus blue. Identify the gas and the ion in the solid, and explain the reaction. The only common gas that turns damp red litmus **blue** is **ammonia**, NH_3, so the solid contains the **ammonium ion**, NH_4^+. The hydroxide ion is the stronger base, so it takes the proton back from the ammonium ion:

$$NH_4^+ + OH^- \rightarrow NH_3 + H_2O$$

This is the standard test for NH_4^+. Always say the litmus is **damp**: the ammonia must dissolve in the water before it can show its basicity, so dry litmus would give no colour change at all.

Exam tips

- N_2 is unreactive because of its **strong triple bond** (very high bond energy) – state this exactly.
- **Ammonia** is a base and a ligand because of its **lone pair**; the ammonium ion forms by a dative bond.
- Explain how **oxides of nitrogen** form (high temperature in engines) and their link to acid rain and photochemical smog.
- Give balanced equations and correct observations for reactions of ammonia and the ammonium ion.

An introduction to AS Level organic chemistry

A-Level Chemistry

Formulas, functional groups and naming



Crude oil is a complex mixture of hydrocarbons – the feedstock for organic chemistry.

A **hydrocarbon** is a compound of only carbon and hydrogen. **Alkanes** are the simplest hydrocarbons and have no functional group.

A **functional group** is the reactive part of a molecule. It decides the physical and chemical properties of the compound, so molecules with the same functional group behave alike (for example the –OH group in alcohols).

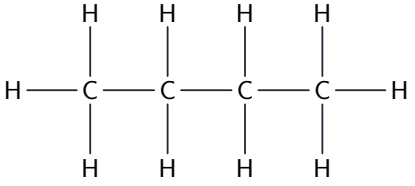
Types of formula

Formula	What it shows
general formula	the pattern for a whole family, e.g. alkanes are C_nH_{2n+2}
molecular formula	the actual number of each atom, e.g. C_4H_{10}
structural formula	the groups in order, e.g. $CH_3CH_2CH_2CH_3$
displayed formula	every atom and every bond drawn out
skeletal formula	lines for bonds; carbons at corners, hydrogens on carbon not shown

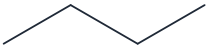
molecular C_4H_{10}

structural $CH_3CH_2CH_2CH_3$

displayed



skeletal



The same molecule (butane) shown four ways – molecular, structural, displayed and skeletal – each hiding more detail than the last

You can read off the **empirical formula** (simplest ratio) from any of these.

Naming

Use systematic **nomenclature** : a stem for the number of carbons (meth-, eth-, prop-, but-, pent-, hex- for 1 to 6), an ending for the functional group, and numbers to show where groups are.

Characteristic organic reactions

Some key terms

- a **homologous series** is a family of compounds with the same functional group and general formula, where successive members differ by CH_2.
- a **saturated** compound has only single C–C bonds; an **unsaturated** compound has a C=C double bond (or a triple bond).

Breaking bonds

A covalent bond can break in two ways:

- **homolytic fission** : the bond splits evenly, one electron to each atom. This makes two **free radicals** (species with an unpaired electron).
- **heterolytic fission** : the bond splits unevenly, both electrons going to one atom. This makes two ions.

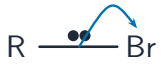
homolytic fission



$Cl\bullet + \bullet Cl$

one electron each
⇒ two **radicals**

heterolytic fission



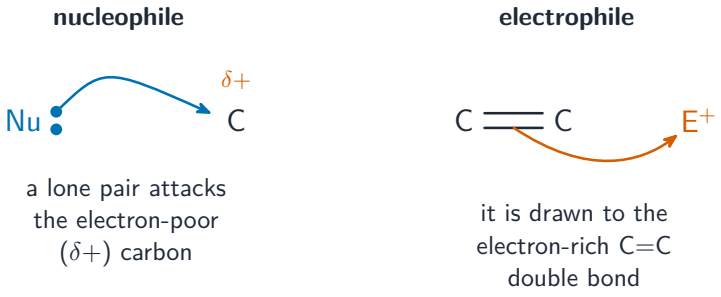
$R^+ + :Br^-$

both electrons to one
⇒ two **ions**

Homolytic fission gives one electron to each atom (two radicals); heterolytic fission gives both electrons to one atom (two ions)

Attacking species

- a **nucleophile** is an electron-pair donor: it uses a lone pair to form a bond to an electron-poor centre. A nucleophile can be neutral, such as ammonia, or negative, such as hydroxide.
- an **electrophile** is an electron-pair acceptor: it accepts a pair from an electron-rich region, such as the pi bond of C=C.



A nucleophile uses its lone pair to attack an electron-poor centre; an electrophile is drawn to an electron-rich one such as a C=C bond

Types of reaction

Reaction	What happens
addition	two molecules join to make one
substitution	one atom or group is swapped for another
elimination	a small molecule is removed, making a double bond
hydrolysis	a molecule is split apart by water
condensation	two molecules join and a small molecule (such as water) is lost

For organic redox, the symbol [O] stands for one oxygen atom from an oxidising agent, and [H] for one hydrogen atom from a reducing agent.

Worked classification. Ethene plus HBr makes bromoethane by addition; bromoethane plus hydroxide replaces Br with OH by substitution; ethanol losing water to make ethene is elimination. Ester plus water giving acid and alcohol is hydrolysis, while acid plus alcohol giving ester and water is condensation. Classify from the structural change and the role of water.

Types of mechanism

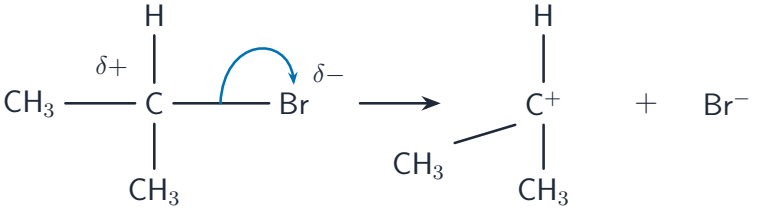
A free-radical reaction happens in three steps: **initiation** (radicals are made), **propagation** (radicals react and make new radicals), and **termination** (two radicals join and stop the chain).

The main mechanisms you meet are **free-radical substitution** (alkanes), **electrophilic addition** (alkenes), **nucleophilic substitution** (halogenoalkanes) and **nucleophilic addition** (carbonyls). In mechanisms, a full **curly arrow** shows a pair of electrons moving; it starts at a bond or a **lone pair** .

Follow the electron pair in a substitution

A full curly arrow starts on a bond or a lone pair and shows **two** electrons moving. A half-headed arrow shows one electron moving in a radical step. Do not start an arrow at a positive charge.

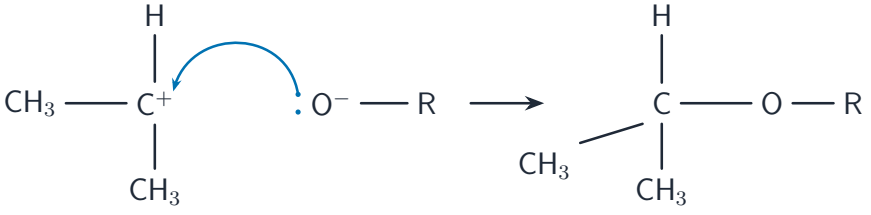
Original application: 9701/24 N25 Q3(b)(ii). The question states an S_N1 mechanism for 2-bromopropane with $(CH_3)_2CHO^-$. Follow the stated mechanism. First show the $\delta+$ carbon and $\delta-$ bromine, then move the C–Br pair to bromine. This gives a carbocation and Br^-.



Bond pair moves to Br; the carbon intermediate has a positive charge.

Full curly arrow from the C–Br bond to Br; the intermediate carbon has a positive charge.

Next, move a lone pair on the nucleophile's oxygen to the positive carbon. A new C–O bond forms. In the figure, R means $CH(CH_3)_2$.



An oxygen lone pair forms the C–O bond. Here R = $CH(CH_3)_2$.

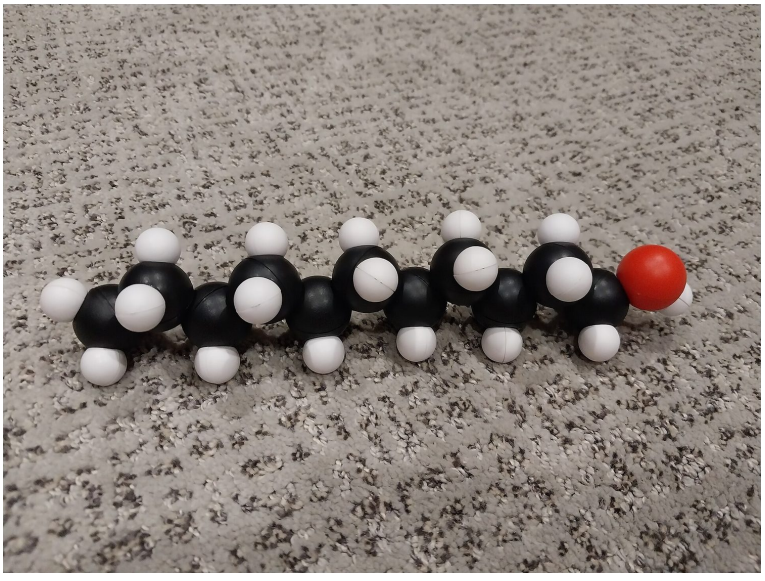
The arrow begins at the lone pair; it shows electron movement, not the movement of an entire molecule.

For the worksheet's 2-bromo-2-methylpropane case, the same two-stage pattern gives a tertiary carbocation, then attack by OH^-. Three alkyl groups help stabilise the positive charge by their electron-releasing effect. A primary carbocation is much less stable. In the worksheet's 1-bromobutane S_N2 case, attack and C–Br breaking occur together in one stage, without a carbocation intermediate. Draw both pair-moving arrows in that stage.

Identify radical stages from what they do

For methane and chlorine under UV light: initiation $Cl_2 \rightarrow 2Cl\cdot$ makes radicals. Propagation $Cl\cdot + CH_4 \rightarrow HCl + \cdot CH_3$, followed by $\cdot CH_3 + Cl_2 \rightarrow CH_3Cl + Cl\cdot$, consumes and regenerates one radical in each step. Termination $\cdot CH_3 + Cl\cdot \rightarrow CH_3Cl$ consumes two radicals without making a new radical.

Shapes of organic molecules



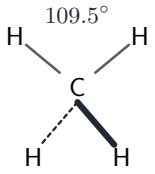
Organic molecules have definite three-dimensional shapes.

Organic molecules can be straight-chained, branched or **cyclic** (in a ring).

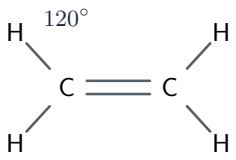
The shape around a carbon depends on its **hybridisation** :

Hybridisation	Bonds	Shape	Angle
sp^3	4 single	tetrahedral	109.5°
sp^2	1 double + 2 single	**planar** (flat)	120°
sp	1 triple (or 2 doubles)	linear	180°

Every single bond is a **sigma bond** σ , made by direct overlap. A double bond is one sigma bond plus one **pi bond** π , made by sideways overlap of p orbitals. Ethene is planar because of its sp^2 carbons.



sp^3: 4 single bonds
tetrahedral

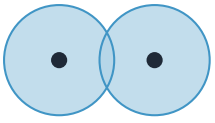


sp^2: 1 double + 2 single
planar (flat)

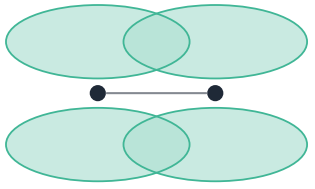


sp: 1 triple bond
linear

The shape at a carbon follows from its hybridisation: sp^3 is tetrahedral (109.5°), sp^2 is planar (120°), sp is linear (180°)



σ bond:
head-on overlap



π bond: sideways overlap
of p orbitals

Sigma density lies on the internuclear axis; pi density is on both sides of it.

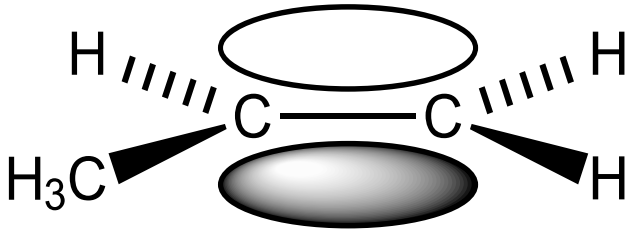
■ Work locally, then count the whole molecule

Original application: 9701/14 N25 Q6. In propyne, CH_3–C ≡ CH, the methyl carbon has four bonding regions: sp^3, tetrahedral, 109.5°. Each triple-bond carbon has two regions: sp, linear, 180°. A multiple bond counts as one region for shape. The molecule therefore contains sp and sp^3 carbons, not sp^2.

For bond counts, each bonded atom pair contributes one σ bond. A double bond adds one π; a triple bond adds two π bonds. Methane has four σ and no π bonds. Ethene has **five** σ (four C–H and one C–C) and **one** π. Ethyne has three σ and two π bonds. Include the C–H bonds when counting the whole molecule.

Original application: 9701/12 N25 Q25. Methanal, $H_2C{=}O$, has three bonding regions around carbon: sp^2, trigonal planar, approximately 120°. All four atoms are in a plane. It has three σ bonds (two C–H and one C–O) plus one π bond. Both original statements are correct: answer A.

Orbital drawing: 9701/24 N25 Q3(a)(iv). The original scheme shows the π overlap above and below the propene framework:



Sideways overlap of parallel p orbitals forms one pi bond with two regions of density; the regions are not separate pi bonds.

Transfer to a non-organic molecule: 9701/12 N25 Q5. The question states that each N in N_2 is sp hybridised. Of its two sp orbitals, one holds a lone pair and one forms a head-on sp–sp σ overlap. The two remaining p orbitals form two π bonds by sideways overlap. The original answer is B.

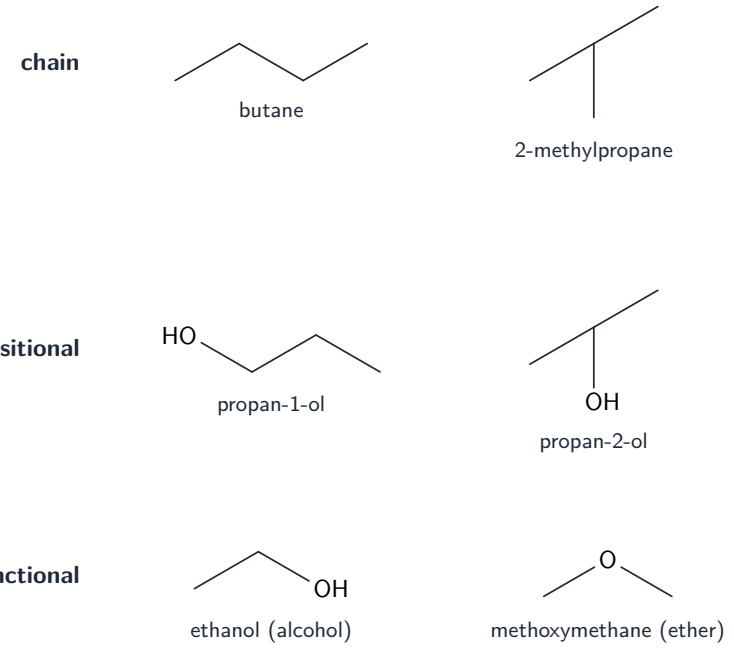
Isomerism

Isomers are compounds with the same molecular formula but a different arrangement of atoms. This is called **isomerism** .

■ Structural isomerism

In **structural isomerism** the atoms are joined in a different order. There are three kinds:

- **chain isomerism** : the carbon chain is branched in different ways.
- **positional isomerism** : the functional group is on a different carbon.
- **functional group isomerism** : the atoms form a different functional group (for example an alcohol and an ether).



The three kinds of structural isomerism – chain, positional and functional-group – each pair sharing the same molecular formula

Stereoisomerism

In **stereoisomerism** the atoms are joined in the same order but point in different directions in space.

- **geometrical isomerism** (cis/trans) happens at a C=C double bond. The pi bond stops the two carbons rotating (**restricted rotation**), so groups are fixed on the same side (**cis**) or opposite sides (**trans**).

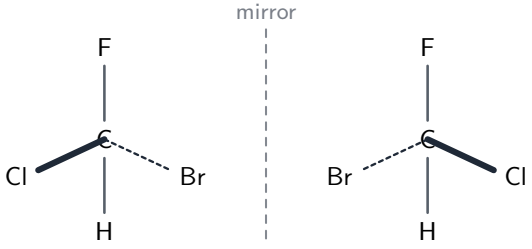


cis: both CH_3 on the same side

trans: CH_3 on opposite sides

Cis–trans isomerism at a C=C bond: the methyl groups are fixed on the same side (cis) or opposite sides (trans) because the bond cannot rotate

- **optical isomerism** happens at a **chiral** carbon – a **chiral centre** is a carbon with four different groups attached. Such a carbon gives two mirror-image forms called **enantiomers** . A molecule may have more than one chiral centre.



enantiomers: mirror images that cannot be superimposed
(the central carbon has 4 different groups)

Optical isomerism: a carbon with four different groups gives two mirror-image forms (enantiomers) that cannot be superimposed

From a molecular formula you can deduce the possible isomers by trying different chains, positions and functional groups.

> **Worked example.** Explain why but-2-ene shows **cis-trans** (E/Z) isomerism but but-1-ene does not. Stereoisomerism at a C=C needs two things: restricted rotation about the double bond (both alkenes have that), **and** two **different** groups on **each** of the two double-bond carbons. In but-2-ene, $CH_3CH{=}CHCH_3$, each double-bond carbon carries an H and a CH_3 - different, so the methyl groups can sit on the same side (**cis** / **Z**) or on opposite sides (**trans** / **E**). In but-1-ene, $CH_2{=}CHCH_2CH_3$, the first carbon carries **two hydrogens** - identical, so swapping them changes nothing and only one form exists. Test **each** double-bond carbon separately: two identical groups on **either** carbon kills the isomerism, however different the other carbon may be.

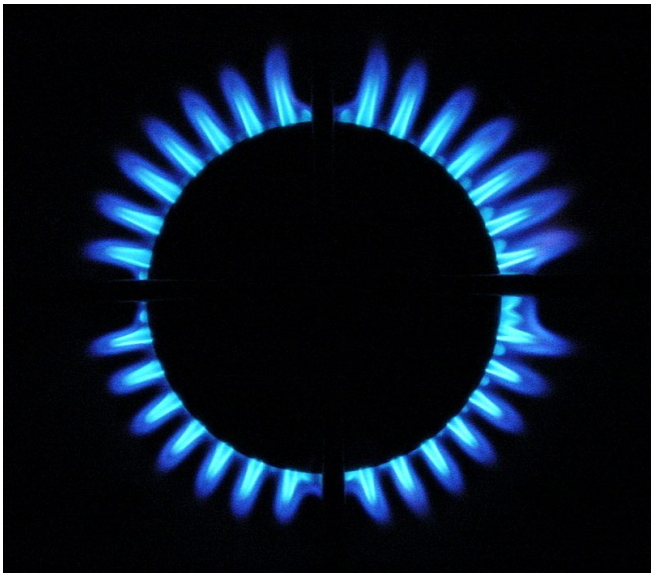
Exam tips

- Know the difference between **general, molecular, structural, displayed and skeletal** formulas – questions ask for a specific one.
- Name systematically: longest chain, lowest locants, substituents alphabetical; a wrong locant loses the mark.
- Classify each reaction by **type and reagent** (addition, substitution, elimination, oxidation).
- **E/Z isomerism** needs restricted rotation about a C = C **and** two different groups on each carbon.

Hydrocarbons

A-Level Chemistry

Alkanes



Natural gas – mainly methane, the simplest alkane – burns on a stove.

Alkanes are saturated hydrocarbons (general formula C_nH_{2n+2}).

Making alkanes

- **hydrogenation** : add hydrogen to an **alkene** , using a Pt or Ni catalyst and heat.
- **cracking** : break a long-chain alkane into shorter ones by heating with Al_2O_3.

Combustion

In **combustion** an alkane burns in oxygen:

$$\text{complete: alkane} + O_2 \rightarrow CO_2 + H_2O$$

$$\text{incomplete: alkane} + \text{less } O_2 \rightarrow CO + \text{soot} + H_2O$$

Complete combustion gives CO2 and water; incomplete also gives CO and soot

- **complete combustion** (plenty of oxygen) gives carbon dioxide and water.

- **incomplete combustion** (not enough oxygen) gives water plus toxic **carbon monoxide** (CO) and soot (carbon).

You are often asked to **write the balanced equation**. Balance it in a fixed order - **carbon first, then hydrogen, and oxygen last** - because oxygen is the only element left on just one side:

$$C_6H_{14} + 9\tfrac{1}{2}\,O_2 \rightarrow 6\,CO_2 + 7\,H_2O$$

The 6 carbons fix $6\,CO_2$; the 14 hydrogens fix $7\,H_2O$; counting the oxygens on the right gives $12+7=19$, so the left needs $19 \div 2 = 9\tfrac{1}{2}$. A **half** of O_2 is perfectly acceptable in this equation - and if the question asks for whole numbers, simply double everything ($2\,C_6H_{14} + 19\,O_2 \rightarrow 12\,CO_2 + 14\,H_2O$).

Free-radical substitution

Alkanes react with chlorine or bromine by **free-radical substitution**, in **ultraviolet light**. For ethane and chlorine the mechanism has three steps:

- **initiation** – UV light splits the halogen into two **free radicals**: $Cl_2 \rightarrow 2\,Cl\cdot$ This kind of break is **homolytic fission**: the bond splits **evenly**, one electron going to each atom, which is what makes two radicals. (The opposite, **heterolytic fission**, sends **both** electrons to one atom and makes a pair of **ions** - that is what happens in the polar mechanisms such as electrophilic addition.) Examiners ask for this word by name, so use it.
- **propagation** – radicals react and make new radicals:

$$Cl\cdot + C_2H_6 \rightarrow C_2H_5\cdot + HCl \qquad C_2H_5\cdot + Cl_2 \rightarrow C_2H_5Cl + Cl\cdot$$

- **termination** – two radicals join, ending the chain: $Cl\cdot + C_2H_5\cdot \rightarrow C_2H_5Cl$

1. initiation

$$Cl_2 \xrightarrow{UV} 2\,Cl^{\bullet}$$

UV light splits Cl–Cl into two radicals

2. propagation

$$Cl^{\bullet} + C_2H_6 \longrightarrow C_2H_5{}^{\bullet} + HCl$$

$$C_2H_5{}^{\bullet} + Cl_2 \longrightarrow C_2H_5Cl + Cl^{\bullet}$$

each radical makes a new radical, so the chain continues

3. termination

$$Cl^{\bullet} + C_2H_5{}^{\bullet} \longrightarrow C_2H_5Cl$$

two radicals join, ending the chain

Free-radical substitution in three steps: initiation makes radicals, propagation carries the chain, and termination ends it

Why cracking is useful, and why alkanes are unreactive

Cracking turns heavy fractions of **crude oil** into more useful, lower-M_r alkanes and alkenes. (A **fraction** is a group of molecules with a similar boiling-point range.)

Alkanes are generally unreactive, especially towards polar reagents. This is because the C–H and C–C bonds are strong and have little **polarity**, so there is no charge to attract an attacking species.

Environmental effects

Burning alkanes in an internal combustion engine gives off carbon monoxide, oxides of nitrogen and unburnt hydrocarbons. A catalytic converter removes these by turning them into harmless gases.

Alkenes



Cracking heavy fractions at a refinery produces alkenes for plastics and fuels.

Alkenes have a C=C double bond (general formula C_nH_{2n}). The double bond is the functional group, so alkenes are reactive.

Making alkenes

- **elimination** of HX from a **halogenoalkane**, using NaOH dissolved in ethanol, with heat.
- **dehydration** of an **alcohol**, using a hot Al_2O_3 catalyst or concentrated sulfuric acid.
- cracking of a longer-chain alkane.

■ Reactions of alkenes

Most reactions are **electrophilic addition** across the double bond:

Reagent and conditions	Product
H_2, Pt/Ni catalyst, heat	alkane
steam (H_2O), H_3PO_4 catalyst	alcohol
a **hydrogen halide** (HX), room temperature	halogenoalkane
a halogen X_2	a di-substituted alkane

There are also two oxidation reactions with acidified $KMnO_4$:

- **cold, dilute** $KMnO_4$ adds two –OH groups to give a **diol** .
- **hot, concentrated** $KMnO_4$ breaks the C=C bond right apart, and what each half turns into tells you exactly where the double bond used to be.

That second reaction is worth learning properly, because it is how the exam asks you to **locate a double bond in a large molecule**. Oxidation is written with [O], meaning "an oxygen atom from the oxidising agent". Cut the molecule at the C=C, then look at **what each of the two carbons was carrying**:

The C=C carbon carries	It becomes
two alkyl groups (=CR_2)	a **ketone** , $R_2C{=}O$
one alkyl and one H (=CHR)	a **carboxylic acid** , RCOOH
two hydrogens (=CH_2)	CO_2 and water

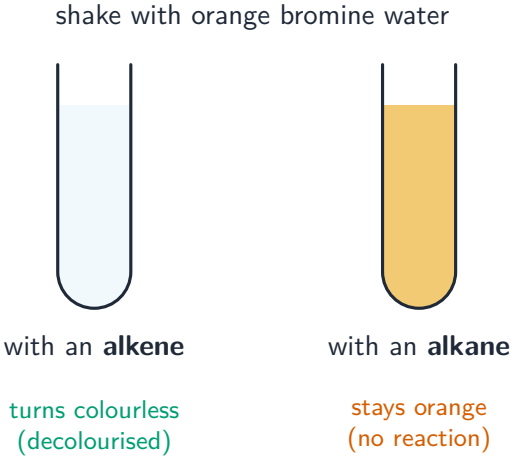
The pattern is simply how many hydrogens that carbon had: none stops at a ketone, one is pushed on to an acid, and two are oxidised all the way to CO_2.

> **Worked example.** Give the products when $(CH_3)_2C{=}CHCH_3$ is heated with hot concentrated acidified $KMnO_4$. Cut at the C=C and take each carbon separately. The **left** carbon carries **two methyl groups** and no hydrogen, so it stops at a **ketone**: $(CH_3)_2C{=}O$, which is propanone. The **right** carbon carries **one methyl and one H**, so it goes on to a **carboxylic acid**: CH_3COOH, ethanoic acid. So:
>
> $$(CH_3)_2C{=}CHCH_3 + 3[O] \rightarrow CH_3COCH_3 + CH_3COOH$$

Now run it **backwards**, which is the way the question is usually set. Given the products, rebuild the alkene by putting the two carbonyl carbons back together as a C=C: a ketone means that carbon had **two alkyl groups**, an acid means **one alkyl and one H**, and CO_2 means the chain **ended** in $=CH_2$. Getting CO_2 is the strongest clue of all - it can only come from a **terminal** double bond.

Test for a C=C bond

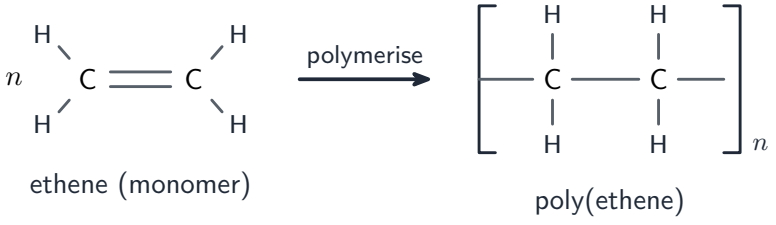
Shake the compound with orange **bromine water** . An alkene **decolourises** it (turns it colourless) by electrophilic addition. An alkane does not.



The bromine-water test: an alkene decolourises the orange bromine water, while an alkane leaves it orange

Addition polymerisation

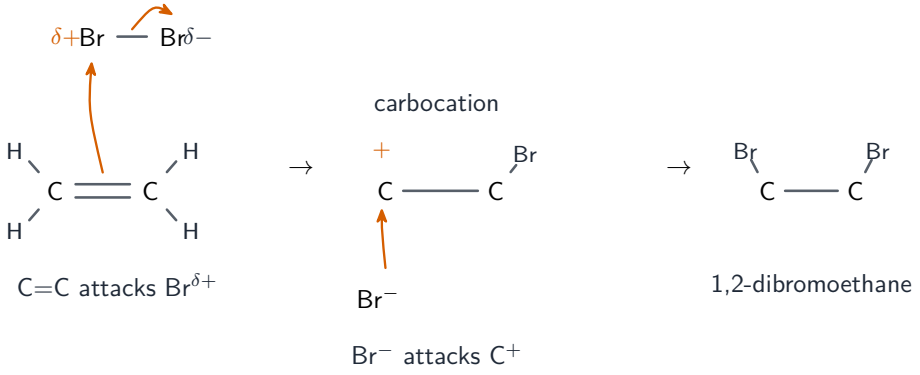
In **addition polymerisation** , many alkene molecules join into one long chain, with no other product. Ethene gives poly(ethene). The long-chain product is a **polymer** .



Addition polymerisation: many ethene molecules open their double bonds and join into the long chain of poly(ethene)

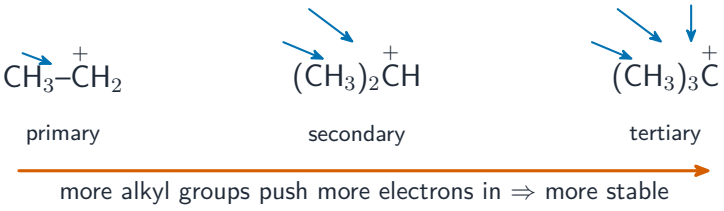
The mechanism and Markovnikov's rule

In electrophilic addition (for example bromine with ethene), the electron-rich C=C attracts the electrophile. This forms a positive intermediate called a **carbocation** , which the negative part then attacks.



Electrophilic addition of bromine to ethene: the C=C attacks $Br^{\delta+}$, a carbocation forms, then Br^- attacks it

Alkyl groups push electrons towards the positive carbon – this is the **inductive effect** . So a carbocation with more alkyl groups is more stable: tertiary is more stable than secondary, which is more stable than primary. When HBr adds to propene, the more stable carbocation forms, so hydrogen adds to the carbon that already has more hydrogens. This pattern is **Markovnikov's rule** .



Markovnikov: HBr adds so the *more stable* carbocation forms

Carbocation stability rises from primary to tertiary as more alkyl groups push electrons in – the basis of Markovnikov's rule

Worked example. Predict the major product when HBr adds to propene, $CH_3CH{=}CH_2$. The H^+ adds first, and it adds in whichever way makes the **more stable carbocation**. Adding the H to the **end** carbon puts the positive charge on the middle carbon, giving a **secondary** carbocation, which is stabilised by electron-releasing alkyl groups on **two** sides. Adding it to the middle carbon would leave a less stable **primary** carbocation. The bromide ion then attacks the secondary carbocation, so the major product is **2-bromopropane**. That is Markovnikov's rule - but quote the **reason** (carbocation stability: tertiary > secondary > primary), because the rule on its own is not the explanation.

Exam tips

- **Alkanes**: free-radical substitution needs **UV light** – show initiation, propagation and termination with the radical dots.
- **Alkenes**: electrophilic addition via a **carbocation**; draw curly arrows from the C = C.
- **Markovnikov**: H adds to the carbon with more H's, via the **more stable carbocation** (alkyl groups are electron-releasing).
- Test for C = C: **bromine water decolourises** (orange → colourless) – state the observation.

Halogen compounds

A-Level Chemistry

Halogenoalkanes

A **halogenoalkane** is an alkane with one or more halogen atoms in place of hydrogen (a C–X bond, where X is a halogen).



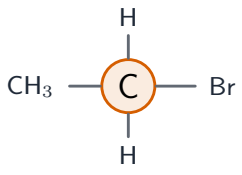
Volatile halogenoalkanes were once widely used as refrigerants and aerosol propellants

Making halogenoalkanes

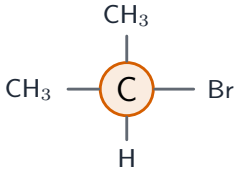
- **free-radical substitution** of an alkane with Cl_2 or Br_2 in ultraviolet light.
- **electrophilic addition** of an alkene with a halogen X_2 or a hydrogen halide HX.
- substitution of an **alcohol**, for example by HX, by PCl_5, by PCl_3 with heat, or by $SOCl_2$.

Three classes

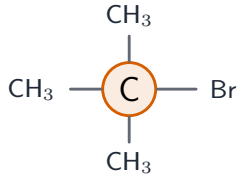
A halogenoalkane is **primary**, **secondary** or **tertiary**, depending on how many carbon atoms are joined to the carbon that holds the halogen (one, two or three).



primary
1 C on the C–Br carbon



secondary
2 C



tertiary
3 C

Primary, secondary and tertiary halogenoalkanes, set by how many carbons are joined to the carbon bearing the halogen

Nucleophilic substitution

The C–X bond is polar, so the carbon is slightly positive. A **nucleophilic substitution** happens when a **nucleophile** (a lone-pair species) attacks that carbon and replaces the halogen.



Many halogenoalkane reactions are carried out by heating the mixture under reflux

Reagent and conditions	Product
NaOH(aq), heat	an alcohol
KCN in ethanol, heat	a **nitrile** (adds one carbon to the chain)
NH_3 in ethanol, heated under pressure	an **amine**



A rotary evaporator removes solvent under reduced pressure – the step after the reaction

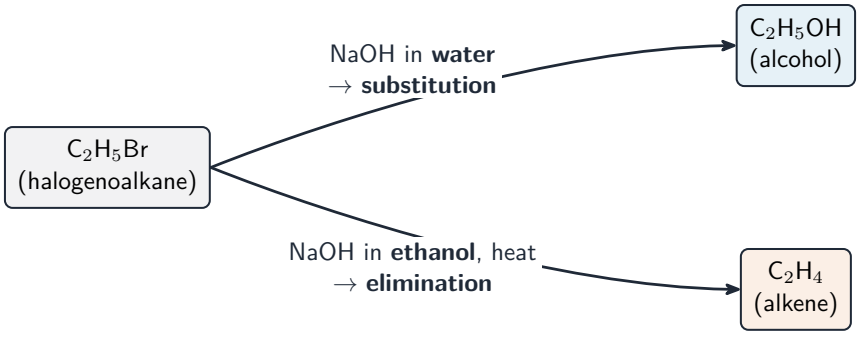
To **identify the halogen**, warm the halogenoalkane with **silver nitrate** in ethanol. A silver halide **precipitate** forms, and its colour shows which halogen is present (white AgCl, cream AgBr, yellow AgI).

Elimination

The same halogenoalkane can instead undergo **elimination** to form an **alkene** . The conditions decide which reaction wins:

- NaOH in **water** → nucleophilic substitution → an alcohol.
- NaOH in **ethanol**, heated → elimination → an alkene.

$$C_2H_5Br + NaOH \rightarrow C_2H_4 + NaBr + H_2O$$



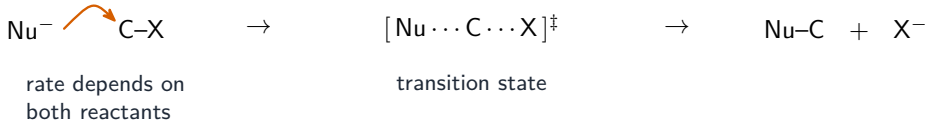
The same halogenoalkane: NaOH in water substitutes to an alcohol, while NaOH in ethanol (with heat) eliminates to an alkene

The S_N1 and S_N2 mechanisms

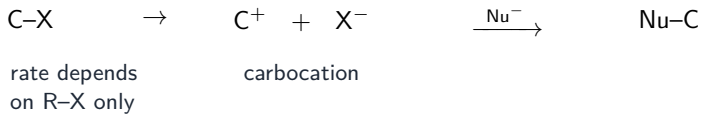
Nucleophilic substitution can follow two routes:

- **S_N2**: one step. The nucleophile attacks at the same time as the halogen leaves, passing through a crowded **transition state** where both are half-bonded. The rate depends on both the halogenoalkane and the nucleophile.
- **S_N1**: two steps. First the C–X bond breaks to give a **carbocation** ; then the nucleophile attacks it. The rate depends only on the halogenoalkane.

S_N2: one step



S_N1: two steps



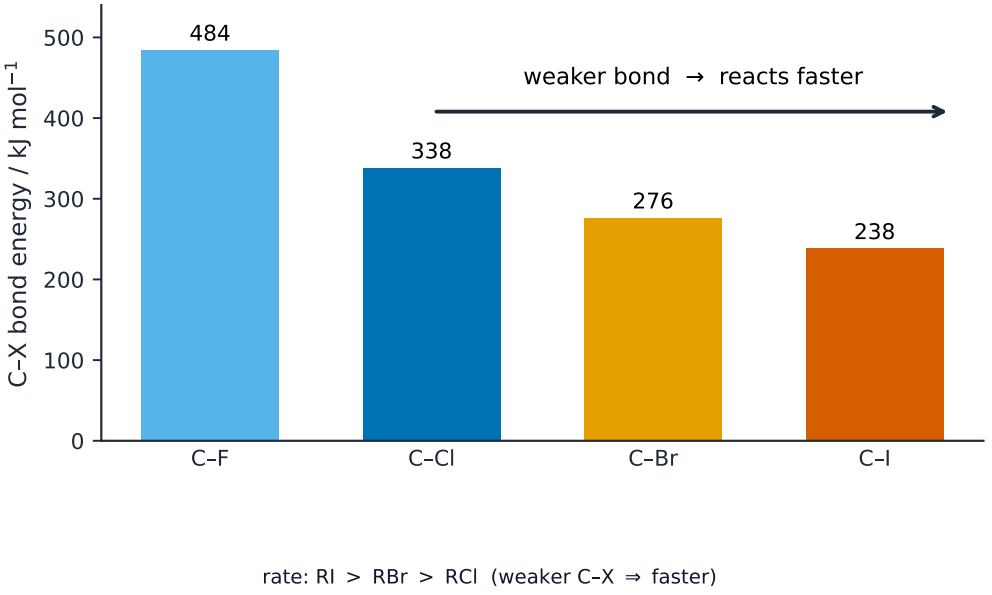
S_N2 is a single step through a crowded transition state; S_N1 is two steps via a carbocation

Alkyl groups push electrons towards the positive carbon (the **inductive effect**), so they stabilise the carbocation. This is why:

- **primary** halogenoalkanes mostly react by S_N2.
- **tertiary** halogenoalkanes mostly react by S_N1 (their carbocation is well stabilised).
- **secondary** halogenoalkanes use a mixture of the two.

Different reactivities

How fast a halogenoalkane reacts depends on the strength of the C–X bond, measured by its **bond energy** . The C–I bond is the weakest, so iodoalkanes react fastest; the C–Cl bond is the strongest of the three, so chloroalkanes react slowest. So when tested with silver nitrate, an iodoalkane gives its precipitate first – its higher **reactivity** comes from the weaker C–X bond.



The weaker the C–X bond, the faster the halogenoalkane reacts – so iodoalkanes react fastest and chloroalkanes slowest

Worked example. Predict the mechanism for the hydrolysis of 1-bromobutane and of 2-bromo-2-methylpropane. Classify the halogenoalkane first. 1-bromobutane is **primary**: the carbon carrying the Br is barely shielded, so the nucleophile can attack the back of it and the mechanism is S_N2 - one step, with the rate depending on **both** the halogenoalkane and the nucleophile. 2-bromo-2-methylpropane is **tertiary**: three bulky methyl groups block that attack, but they also stabilise the carbocation formed once the Br leaves, so it goes S_N1 - two steps, with the rate depending on the halogenoalkane **only**. Decide from the **class** (primary → S_N2, tertiary → S_N1), and note that the tertiary one hydrolyses **faster** despite being the more crowded.

Exam tips

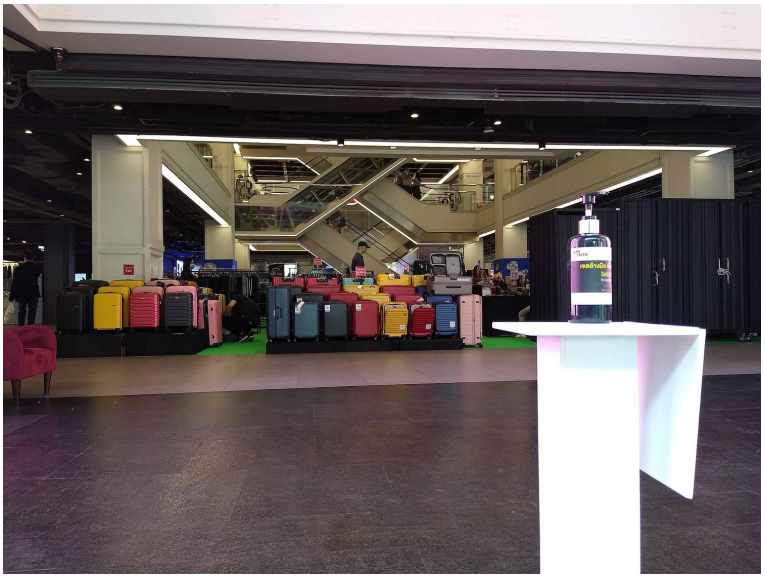
- Draw **nucleophilic substitution** with a curly arrow from the nucleophile's lone pair to the $\delta+$ carbon and one from the C-X bond.
- Match mechanism to class: S_N1 (tertiary, carbocation, two steps) vs S_N2 (primary, one step, transition state).
- Reactivity is set by **bond enthalpy**: C-I is weakest, so iodoalkanes react fastest (not electronegativity).
- **Elimination** (hot, ethanolic KOH) vs **substitution** (warm, aqueous KOH) – the conditions decide the product; state them.

Hydroxy compounds

A-Level Chemistry

Alcohols

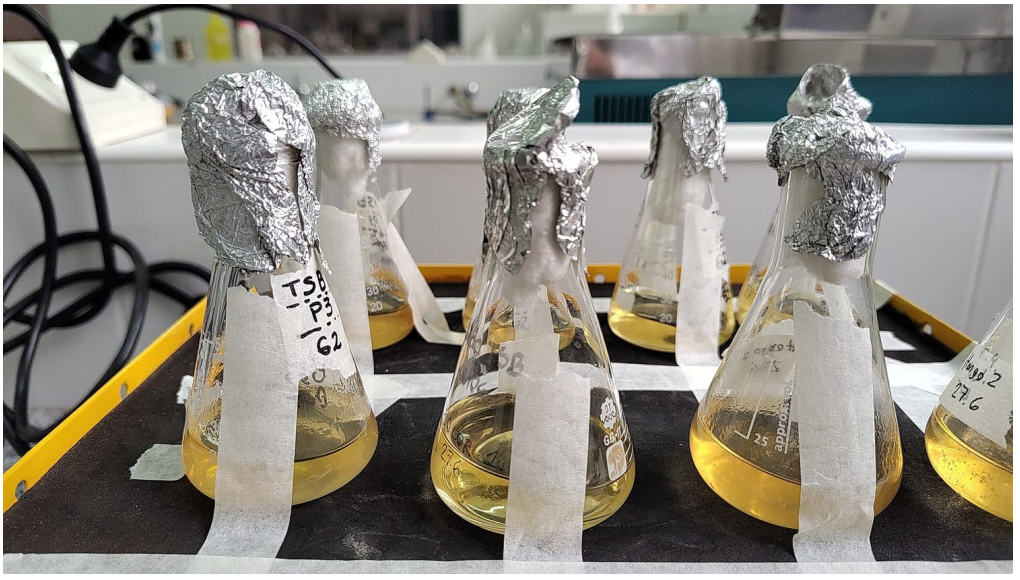
An **alcohol** has the –OH (hydroxyl) functional group.



Ethanol, the alcohol in hand sanitiser, kills microbes

Making alcohols

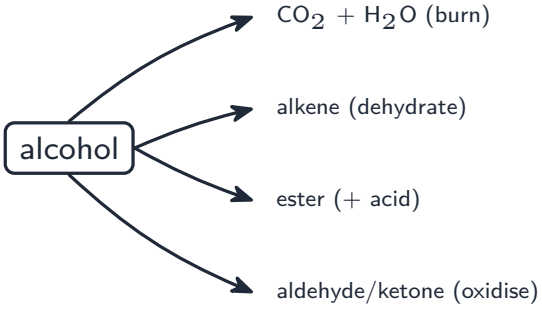
Method	Reagents and conditions
addition of steam to an alkene	$H_2O(g)$, H_3PO_4 catalyst (**electrophilic addition**)
an **alkene** with cold dilute $KMnO_4$	gives a **diol** (two –OH groups)
substitution of a **halogenoalkane**	NaOH(aq), heat
reduction of an **aldehyde** or **ketone**	$NaBH_4$ or $LiAlH_4$
reduction of a **carboxylic acid**	$LiAlH_4$
hydrolysis of an **ester**	dilute acid or alkali, heat



Industrially, ethanol is also made by fermenting sugar; fractional distillation then separates it from the water

Reactions of alcohols

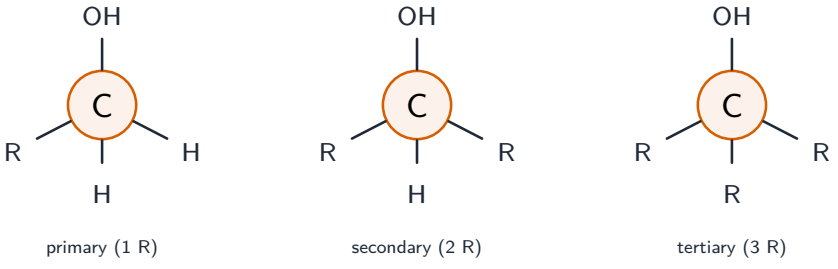
- **combustion**: alcohols burn in oxygen to give carbon dioxide and water.
- **substitution** to a halogenoalkane, for example with HX, PCl_5, PCl_3 and heat, or $SOCl_2$.
- **with sodium**: alcohols react with sodium metal to give hydrogen and a sodium alkoxide – like water, but more slowly.
- **oxidation** with acidified $K_2Cr_2O_7$ (or $KMnO_4$). The product depends on the class of alcohol (see below).
- **dehydration** to an alkene, using a hot Al_2O_3 catalyst or concentrated acid.
- **ester formation**: an alcohol reacts with a carboxylic acid (with concentrated H_2SO_4 catalyst) to make an ester.



An alcohol can burn, dehydrate, esterify or be oxidised

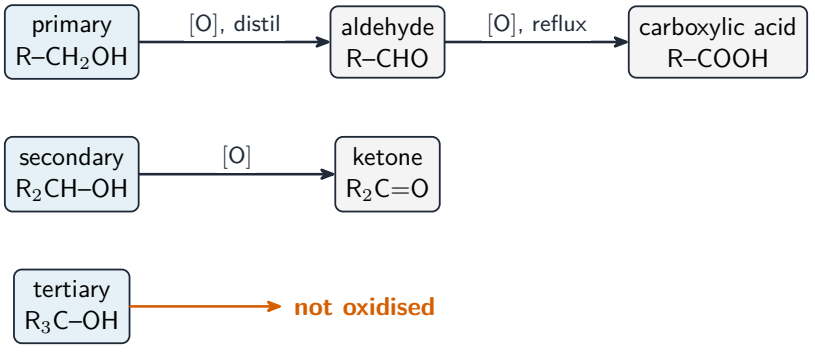
Three classes and how oxidation tells them apart

An alcohol is **primary** , **secondary** or **tertiary** , depending on how many carbons are joined to the carbon holding the –OH. Some molecules have more than one –OH group.



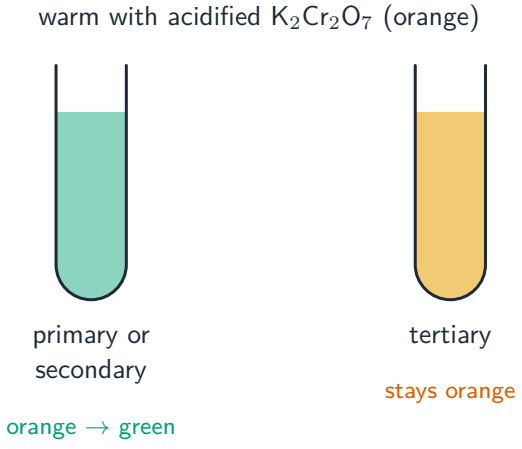
The class is set by the carbon holding the –OH: one R group makes it primary, two secondary, three tertiary – which decides how it oxidises

Class	Oxidation product
primary	**aldehyde** (by **distillation**), then **carboxylic acid** (by **reflux**)
secondary	a **ketone**
tertiary	not oxidised



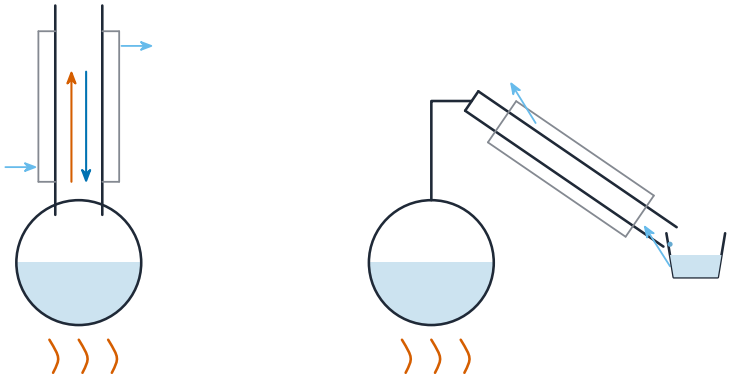
Oxidation by class: a primary alcohol gives an aldehyde then a carboxylic acid, a secondary gives a ketone, a tertiary is not oxidised

In a quick test, acidified $K_2Cr_2O_7$ turns from **orange to green** with a primary or secondary alcohol, but stays orange with a tertiary alcohol.

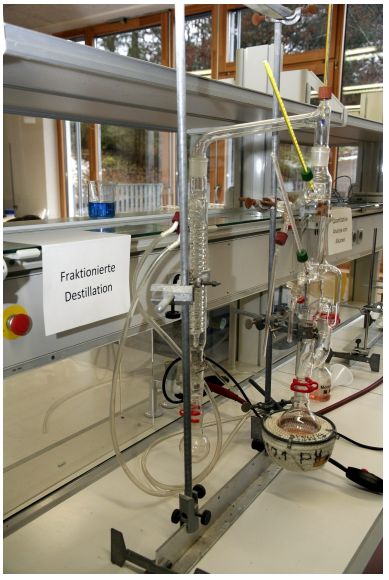


Acidified dichromate turns orange to green with a primary or secondary alcohol, but stays orange with a tertiary alcohol

- **distillation** removes the aldehyde as it forms, before it can be oxidised further.
- **reflux** keeps boiling the mixture and returning the vapour, so the alcohol is fully oxidised to the carboxylic acid.



Distillation removes the aldehyde as it forms; reflux keeps boiling and returning the vapour, fully oxidising to the acid



A real distillation set-up: the vapour boils off, cools in the condenser and is collected – the same idea separates ethanol from a fermented mixture

The iodoform test

If you warm an alcohol that contains the $CH_3CH(OH)-$ group with alkaline aqueous iodine, you get a pale yellow precipitate of **tri-iodomethane** (CHI_3) and the ion RCO_2^-. This is a useful test for that group.

Acidity of alcohols

The –OH group makes alcohols very weakly acidic: they can lose the H^+ to form an RO^- ion. But their **acidity** is **lower** than that of water. This is because the alkyl group pushes electron density onto the oxygen, which makes the RO^- ion less stable, so the alcohol holds onto its H^+ more tightly.

> **Worked example.** Three unlabelled bottles hold butan-1-ol, butan-2-ol and 2-methylpropan-2-ol. How does warming each with acidified $K_2Cr_2O_7$ tell them apart? What matters is how many hydrogens sit on the **carbon carrying the OH**. Butan-1-ol is **primary** (two such hydrogens): the orange dichromate turns **green**, and by distilling you collect an **aldehyde** (butanal), or by refluxing you get the **carboxylic acid** (butanoic acid). Butan-2-ol is **secondary** (one such hydrogen): also **green**, but the product is a **ketone** (butanone), which will not oxidise further. 2-methylpropan-2-ol is **tertiary** (**no** such hydrogen): there is nothing to remove, so the dichromate stays **orange**. The colour only separates the tertiary from the other two - to split primary from secondary you must identify the **product** (an aldehyde gives a silver mirror with Tollens', a ketone does not).

Exam tips

- Classify the alcohol as **primary, secondary or tertiary** first – it decides the oxidation product.
- **Oxidation** with acidified $K_2Cr_2O_7$ (orange → green): primary → aldehyde (distil) → acid (reflux); secondary → ketone; tertiary → no reaction.
- Distinguish the conditions for **aldehyde vs acid** from a primary alcohol (distillation vs reflux).
- Name esters correctly (the acid part comes second, ending "-oate").

Carbonyl compounds

A-Level Chemistry

Aldehydes and ketones

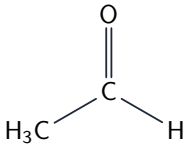
Aldehydes and ketones are **carbonyl compounds** – they contain the C=O **carbonyl** group.

- in an **aldehyde** the carbonyl carbon is on the end of the chain (it also carries an H), written –CHO.
- in a **ketone** the carbonyl carbon is in the middle, between two other carbons.



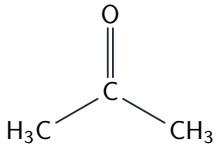
Propanone (acetone), the simplest ketone, is the solvent in nail-polish remover

aldehyde



C=O carbon on the end
(also carries an H): –CHO

ketone



C=O carbon in the middle,
between two carbons

Both have the C=O carbonyl group: in an aldehyde it is on the end of the chain (–CHO), in a ketone it is in the middle

Making aldehydes and ketones

Both are made by the **oxidation** of an **alcohol** with acidified $K_2Cr_2O_7$ or $KMnO_4$:

- a **primary** alcohol, with **distillation**, gives an aldehyde.
- a **secondary** alcohol gives a ketone.

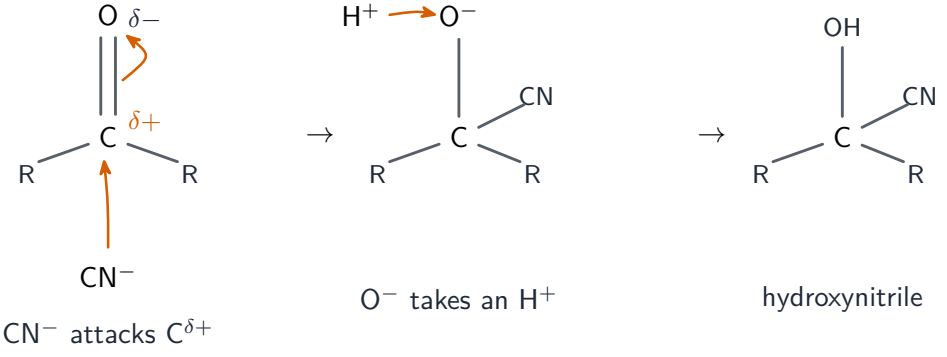
Reactions

- **reduction** with $NaBH_4$ or $LiAlH_4$ turns a carbonyl compound back into an alcohol (an aldehyde gives a primary alcohol; a ketone gives a secondary alcohol).
- reaction with **hydrogen cyanide** (HCN), with KCN as catalyst and heat, adds H and CN across the C=O to make a **hydroxynitrile**. This adds one carbon to the chain.

The mechanism: nucleophilic addition

The reaction with HCN is a **nucleophilic addition**. The carbonyl carbon is slightly positive (oxygen pulls the electrons away). So:

1. the CN^- ion (a nucleophile) attacks the slightly positive carbon.
2. this breaks the C=O double bond, leaving a negative oxygen (O^-).
3. the O^- takes an H^+ (from HCN) to finish the hydroxynitrile.



Nucleophilic addition of HCN: CN^- attacks the $\delta+$ carbonyl carbon, the C=O breaks to O^-, then O^- takes an H^+

Tests for carbonyl compounds

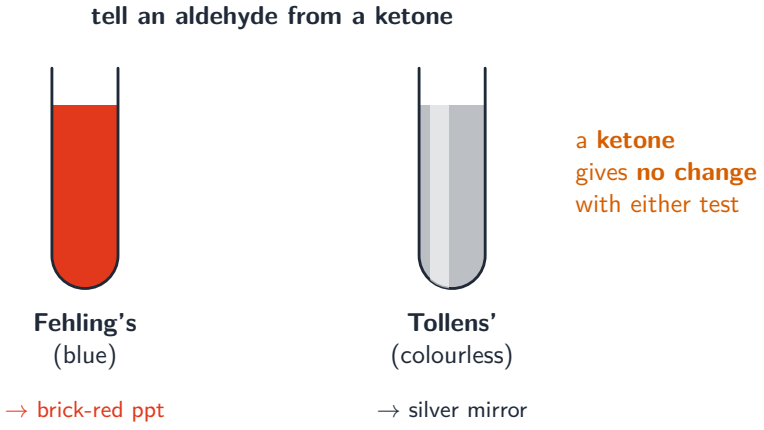
Detecting any carbonyl

Add **2,4-DNPH** reagent (2,4-dinitrophenylhydrazine). An orange precipitate confirms that the compound is an aldehyde or a ketone.

Telling an aldehyde from a ketone

Aldehydes are easily oxidised to carboxylic acids, but ketones are not. Two tests use this difference:

Test	Aldehyde	Ketone
Fehling's reagent (blue solution)	turns to a brick-red precipitate	no change
Tollens' reagent (colourless)	gives a silver mirror	no change



Telling an aldehyde from a ketone: an aldehyde gives a brick-red precipitate with Fehling's and a silver mirror with Tollens'; a ketone gives no change



A positive Tollens' test: an aldehyde coats the tube with a shiny silver mirror

■ The iodoform test

If the compound has the CH_3CO- group, warming it with alkaline aqueous iodine gives a pale yellow precipitate of **tri-iodomethane** (CHI_3) and the ion RCO_2^-.

> **Worked example.** A liquid gives an orange precipitate with 2,4-DNPH, gives **no** silver mirror with Tollens' reagent, and gives a yellow precipitate with alkaline aqueous iodine. Identify it. Take the tests one at a time, using each for exactly what it proves. The orange precipitate with **2,4-DNPH** proves a **carbonyl** group is present - an aldehyde or a ketone, nothing else. **No** silver mirror with Tollens' rules out an **aldehyde**, so it must be a **ketone**. The yellow precipitate in the **iodoform** test proves a CH_3CO- group next to the carbonyl. The simplest compound satisfying all three is **propanone**, CH_3COCH_3. Keep the roles straight: 2,4-DNPH finds **any** carbonyl, Tollens' separates aldehyde from ketone, and iodoform detects the **methyl group beside** the C=O.

Exam tips

- **2,4-DNPH** gives an orange precipitate with **any** carbonyl – it tests for C = O.
- **Tollens' (silver mirror)** and **Fehling's (brick-red)** are positive for **aldehydes only** – this is how you tell aldehydes from ketones.
- Nucleophilic addition of HCN adds **one carbon** ($\rightarrow$ hydroxynitrile); show the curly arrows.
- The **iodoform (tri-iodomethane)** test is positive for CH_3CO- groups.

Carboxylic acids and derivatives

A-Level Chemistry

Carboxylic acids

A **carboxylic acid** has the –COOH (carboxyl) functional group. It is a weak acid.



Vinegar is a dilute solution of ethanoic acid, a carboxylic acid

Making carboxylic acids

- **oxidation** of a primary **alcohol** or an **aldehyde** with acidified $K_2Cr_2O_7$ or $KMnO_4$, with **reflux** (so it is fully oxidised).
- **hydrolysis** of a **nitrile** with dilute acid or alkali, then acidifying.
- hydrolysis of an **ester** with dilute acid or alkali and heat, then acidifying.

Reactions of carboxylic acids

These reactions all show that carboxylic acids are acids:

- **with a reactive metal** (a **redox** reaction): gives a **salt** and hydrogen.

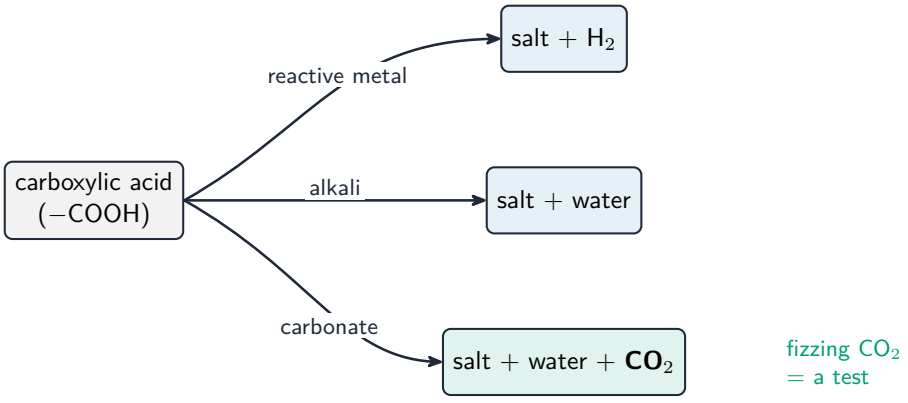
$$2CH_3COOH + Mg \rightarrow (CH_3COO)_2Mg + H_2$$

- **with an alkali** (a **neutralisation**): gives a salt and water.

$$CH_3COOH + NaOH \rightarrow CH_3COONa + H_2O$$

- **with a carbonate** : gives a salt, water and carbon dioxide. The fizzing of CO_2 is a test for a carboxylic acid.

$$2CH_3COOH + Na_2CO_3 \rightarrow 2CH_3COONa + H_2O + CO_2$$



Carboxylic acids behave as acids: a salt with a metal (+ H_2), with an alkali (+ water), or with a carbonate (+ water + CO_2 – the fizz is a test)

Two reactions change the functional group:

- **esterification** with an alcohol (concentrated H_2SO_4 catalyst) gives an ester.
- **reduction** by $LiAlH_4$ gives a primary alcohol.

Esters

An ester has the –COO– group. It often smells sweet or fruity.



Esters give many fruits and perfumes their sweet smell

Making esters

An ester forms in a **condensation** reaction between an alcohol and a carboxylic acid, with concentrated H_2SO_4 as catalyst. A water molecule is lost, and the reaction is reversible:

$$CH_3COOH + C_2H_5OH \rightleftharpoons CH_3COOC_2H_5 + H_2O$$

$$CH_3COOH \quad + \quad HOC_2H_5 \quad \rightleftharpoons \quad CH_3CO\ O\ C_2H_5 \quad + \quad H_2O$$

the –OH (acid) and –H (alcohol) leave as water; the C–O–C ester link forms

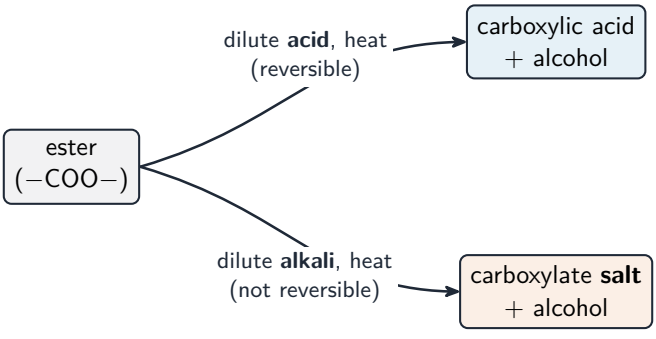
(concentrated H_2SO_4 catalyst; reversible)

Esterification is a condensation: the –OH from the acid and the –H from the alcohol leave as water, forming the C–O–C ester link

Hydrolysis of esters

Hydrolysis splits the ester back apart. The conditions change the products:

- **dilute acid** and heat: reversible. Gives back the carboxylic acid and the alcohol.
- **dilute alkali** and heat: not reversible. Gives the alcohol and the **salt** of the carboxylic acid (the carboxylate ion).



Hydrolysing an ester: dilute acid (reversible) gives the carboxylic acid and alcohol; dilute alkali (not reversible) gives the carboxylate salt and alcohol

Worked example. Name the ester made from ethanol and propanoic acid, and say which part comes from which. In an ester name the **alcohol** gives the first word (the alkyl part) and the **acid** gives the second (the -oate part). Ethanol supplies C_2H_5- and propanoic acid supplies CH_3CH_2COO-, so the ester is **ethyl propanoate**, $CH_3CH_2COOC_2H_5$. It is made by refluxing the two with a concentrated H_2SO_4 catalyst, and the reaction is **reversible**, so the yield is never complete. The trap is naming it backwards: **propyl ethanoate** is a completely different ester (made from propan-1-ol and ethanoic acid). Alcohol first, acid second.

Exam tips

- Carboxylic acids react with carbonates to give **CO_2** (fizzing) – this distinguishes them from phenols.
- Explain acid strength via the **delocalised carboxylate ion**; electron-withdrawing groups (e.g. Cl) increase strength.
- **Esterification** (with concentrated H_2SO_4) is reversible; name esters as *alkyl alkanoate*.
- Know the products of **acid vs alkaline** hydrolysis of an ester (they differ).

Nitrogen compounds

A-Level Chemistry

Primary amines



Dyeing cloth: azo and amine chemistry is what puts fast, bright colour into fibres

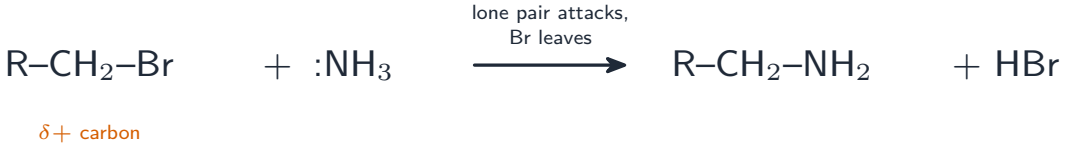
An **amine** has an $-NH_2$ group (the nitrogen replaces a hydrogen of ammonia).

Making a primary amine

Heat a **halogenoalkane** with ammonia dissolved in ethanol, under pressure:

$$C_2H_5Br + NH_3 \rightarrow C_2H_5NH_2 + HBr$$

This is a **nucleophilic substitution**: the lone pair on the nitrogen of ammonia attacks the slightly positive carbon and pushes out the halogen. You use an excess of ammonia, or the amine made can react again.



Making a primary amine: ammonia's lone pair attacks the δ+ carbon and pushes out the halogen – a nucleophilic substitution

Nitriles and hydroxynitriles

Making a nitrile

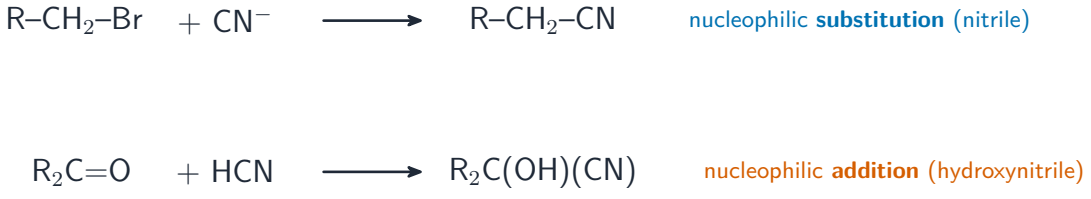
Heat a halogenoalkane with potassium cyanide (KCN) in ethanol:

$$C_2H_5Br + KCN \rightarrow C_2H_5CN + KBr$$

This is also a nucleophilic substitution, with the CN^- ion as the nucleophile. It is useful because it **adds one carbon** to the chain. The product is a **nitrile** .

Making a hydroxynitrile

Add HCN (with KCN as catalyst, and heat) to an **aldehyde** or **ketone** . The H and CN add across the C=O bond to give a **hydroxynitrile** . The reagent is **hydrogen cyanide** , and the mechanism is **nucleophilic addition** .



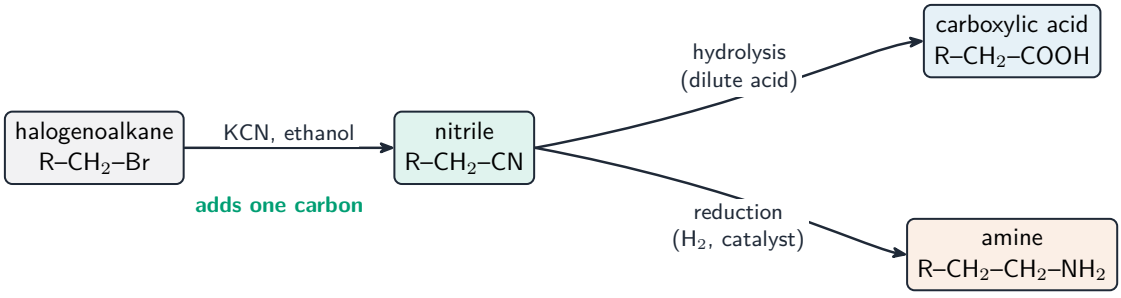
Two ways cyanide adds a carbon: KCN with a halogenoalkane substitutes to a nitrile; HCN with a carbonyl adds to a hydroxynitrile

Hydrolysis of nitriles

Warm a nitrile with dilute acid (or dilute alkali, then acidify). This **hydrolysis** turns the –CN group into a –COOH group, giving a **carboxylic acid** :

$$CH_3CN + 2H_2O + HCl \rightarrow CH_3COOH + NH_4Cl$$

A nitrile can also be **reduced** by hydrogen and a catalyst to form an amine, which is the **reduction** route to a longer-chain amine.



Nitriles are a useful hub: KCN adds a carbon to make the nitrile, which then hydrolyses to a carboxylic acid or reduces to an amine



Reducing a nitrile gives an amine; amines and diacids link up to make polyamides such as nylon – the fibre first made famous in stockings

Worked example. Starting from bromoethane, make propanoic acid. Compare the carbons first: bromoethane has 2, propanoic acid has 3, so a carbon must be **added** - and the KCN step is the reaction that does it. Step 1: warm bromoethane with **ethanolic KCN**; nucleophilic substitution gives propanenitrile, CH_3CH_2CN, which now has 3 carbons because the CN carbon joins the chain. Step 2: reflux the nitrile with **dilute HCl**; hydrolysis gives propanoic acid. Count the carbons **before** you plan: whenever the target has exactly one more than the starting material, the nitrile route is almost always the intended answer, and remember the CN carbon is part of the new chain.

Exam tips

- **Amines are bases** (the N lone pair accepts H^+); aliphatic amines are stronger bases than ammonia.
- Making amines: **reduce a nitrile**, or react a halogenoalkane with excess ammonia.
- **KCN adds one carbon** (halogenoalkane → nitrile) – track the carbon count carefully in synthesis routes.
- State reagents and conditions precisely for each conversion.

Polymerisation

A-Level Chemistry

Addition polymerisation

In **addition polymerisation** , many small molecules join into one very long chain, with no other product made.

Each small molecule is a **monomer** . It must be unsaturated – it has a C=C double bond. The double bond opens up so that the monomers can link together. The long chain that forms is the **polymer** .

many monomers (each has a C=C)

$CH_2{=}CH_2$ $CH_2{=}CH_2$ $CH_2{=}CH_2$

↓ double bonds open and join

$-CH_2-CH_2-CH_2-CH_2-CH_2-CH_2-$

one long polymer chain (single bonds)

Each monomer's double bond opens, and the monomers join end to end into one long chain – with no other product made



Poly(ethene), a common addition polymer, is supplied as tiny pellets a few millimetres across; these are later melted and moulded into bottles, bags and other products

Repeat units

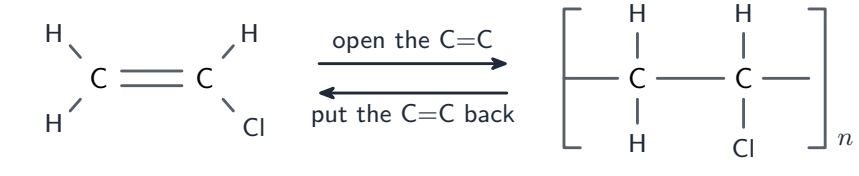
The **repeat unit** is the small part that is copied again and again along the chain. To find it, take the monomer, change the C=C to a single C–C, and draw bonds going out at each end.

- **poly(ethene)** is made from ethene:

$$n\,\mathrm{CH_2{=}CH_2} \rightarrow -(\mathrm{CH_2{-}CH_2})_n-$$

- **poly(chloroethene)** (PVC) is made from chloroethene:

$$n\,\mathrm{CH_2{=}CHCl} \rightarrow -(\mathrm{CH_2{-}CHCl})_n-$$



chloroethene (monomer) poly(chloroethene), PVC

Finding the repeat unit: change the monomer's C=C to a single bond and draw bonds out at each end; reverse it to find the monomer

■ Finding the monomer

To go the other way, look at one repeat unit of the polymer, and put the C=C double bond back in. That gives you the monomer.

> **Worked example.** A polymer has the repeat unit $-(CH_2-CH(CH_3))_n-$. Identify the monomer and name the polymer. Reverse the rule you used to build it: rub out the bonds sticking out at each end, and turn the single C-C **in the backbone** back into a **C=C**. That gives $CH_2=CH(CH_3)$, which is propene - so the polymer is **poly(propene)**. Two checks catch most errors: the monomer must have the **same molecular formula** as the repeat unit (addition polymerisation adds nothing and loses nothing), and the double bond goes back into the **backbone**, never into the side group.

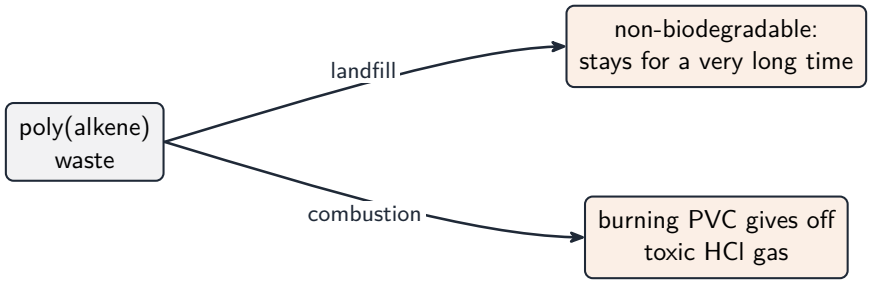
The problem of disposal

Poly(alkene)s are very hard to get rid of:

- they are **non-biodegradable** – microbes cannot break them down, so they stay in the ground for a very long time.
- their **combustion** (burning) can release harmful gases. For example, burning PVC gives off toxic hydrogen chloride.



Most addition polymers are non-biodegradable, so they build up as waste in the environment



Poly(alkene) waste is hard to dispose of: it is non-biodegradable, and burning PVC releases toxic hydrogen chloride

Exam tips

- Draw the **repeat unit** in brackets with the two bonds crossing them and n outside; keep the backbone carbons.
- Deduce the **monomer from a polymer** (and vice versa) – a very common question.
- Addition polymers are **inert and non-biodegradable** – link to landfill, recycling and incineration issues.
- Do not confuse with condensation (no small molecule is lost in addition polymerisation).

Organic synthesis

A-Level Chemistry

This topic does not add new reactions. Instead it asks you to **join up** the reactions you already know, so you can build a target molecule in several steps.



Medicines are built up from simple starting materials by multi-step organic synthesis

Identifying functional groups

A molecule may have more than one **functional group** . Use the test reactions from the syllabus to identify each one, and then predict how the molecule will behave. For example:

- decolourises bromine water → a C=C double bond (an **alkene**).
- gives a precipitate with silver nitrate → a **halogenoalkane** .
- orange $K_2Cr_2O_7$ turns green → a primary or secondary **alcohol** .
- orange precipitate with 2,4-DNPH → an **aldehyde** or **ketone** .
- fizzes with a carbonate → a **carboxylic acid** .

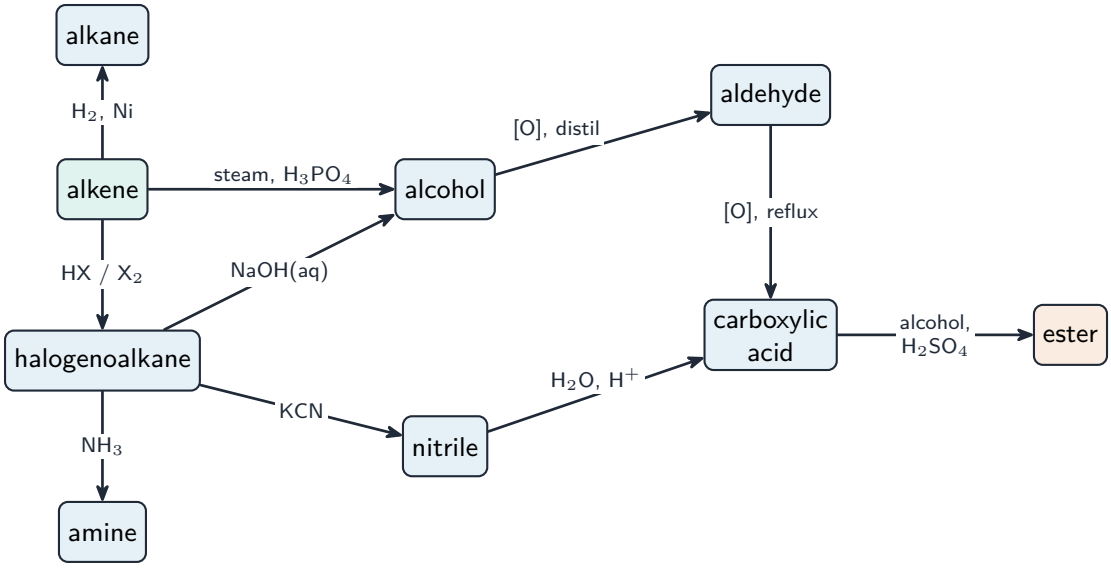
add this	you see	functional group
bromine water	orange → colourless	C=C (alkene)
silver nitrate	a precipitate forms	halogenoalkane
acidified $K_2Cr_2O_7$	orange → green	1°/2° alcohol
2,4-DNPH	orange precipitate	aldehyde / ketone
a carbonate	fizzes (CO_2 given off)	carboxylic acid

The standard identification tests: each reagent gives a characteristic observation that points to one functional group

A map of the AS reactions

Each row turns one functional group into another. Learn it as a map you can travel around:

Start	Reagent and conditions	Product
alkene	H_2, Ni	alkane
alkene	HX, or X_2	halogenoalkane
alkene	steam, H_3PO_4	alcohol
halogenoalkane	NaOH(aq), heat	alcohol
halogenoalkane	KCN in ethanol, heat	a **nitrile** (adds one carbon)
halogenoalkane	NH_3 in ethanol, pressure	an **amine**
alcohol	$K_2Cr_2O_7$, distil / reflux	aldehyde / carboxylic acid
alcohol	concentrated acid, heat	alkene
aldehyde or ketone	$NaBH_4$	alcohol
aldehyde or ketone	HCN, KCN	hydroxynitrile
nitrile	dilute acid, heat	carboxylic acid
carboxylic acid + alcohol	concentrated H_2SO_4	an **ester**



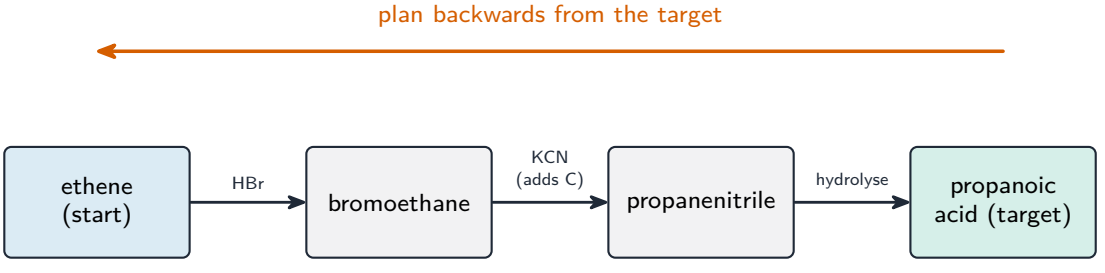
A map of the AS reactions: each arrow turns one functional group into another. Work backwards from your target to plan a route

Planning a multi-step route

To devise a **synthetic route** :

1. compare the target with the starting material – what has changed (the functional group, the number of carbons)?
2. work **backwards** from the target: which single reaction could make it, and from what?
3. repeat until you reach the starting material.
4. write each step with its **reagent** and conditions.

If you need to add a carbon, the KCN step is the key – it is the only AS reaction that lengthens the chain.



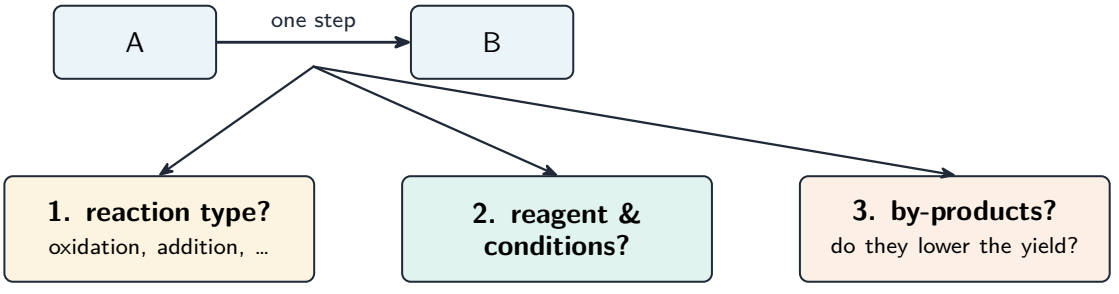
Plan a route by working backwards from the target, one reaction at a time, until you reach the starting material

Analysing a route

When you are given a route, for each step state the **type of reaction** (such as **oxidation** , **reduction** , substitution, addition or elimination) and the reagent used. Also think about possible **by-products** – for example, making an amine from a halogenoalkane also gives a mixture of further-substituted amines, so the yield of the simple amine is low.



A separating funnel splits two immiscible layers – how the organic product is recovered from the reaction mixture



e.g. a halogenoalkane → amine also makes further-substituted amines, so the simple-amine yield is low

For every step in a given route, ask its reaction type, reagent and by-products

name the reaction type at each step

oxidation	loses electrons / gains O or loses H (e.g. alcohol → acid)
reduction	gains electrons / loses O or gains H
substitution	one atom or group swaps for another
addition	two molecules join; a C=C double bond opens
elimination	a small molecule is removed; a C=C forms

also give the **reagent** for each step, and watch for **by-products** (e.g. amines over-substitute)

Name the reaction type at each step: oxidation, reduction, substitution, addition or elimination

Worked example. Devise a route from propene to propanone, CH_3COCH_3. Work **backwards** from the target. A ketone comes from oxidising a **secondary** alcohol, so the step before propanone is propan-2-ol with acidified $K_2Cr_2O_7$ under reflux. Propan-2-ol comes from propene by adding **steam** over an H_3PO_4 catalyst - and Markovnikov's rule conveniently puts the OH on the **middle** carbon, which is exactly the secondary alcohol needed. So the route is: propene, then steam with H_3PO_4, giving propan-2-ol; then acidified $K_2Cr_2O_7$ under reflux, giving propanone. Give a **reagent and its conditions** on every arrow: a route with the right intermediates but no reagents scores very little.

Exam tips

- Learn the **reagents and conditions** for each conversion – that is exactly what synthesis questions test.
- Plan multi-step routes by functional group and watch the **carbon count** (KCN adds one carbon).
- Choose the **shortest valid route** and state every reagent and condition.

Analytical techniques

A-Level Chemistry

Infrared spectroscopy

Infrared spectroscopy helps you find the **functional group** in a molecule. Each kind of bond soaks up (absorbs) infrared radiation at its own range of frequencies. Where the bond shows strong **absorption**, the spectrum has a dip.

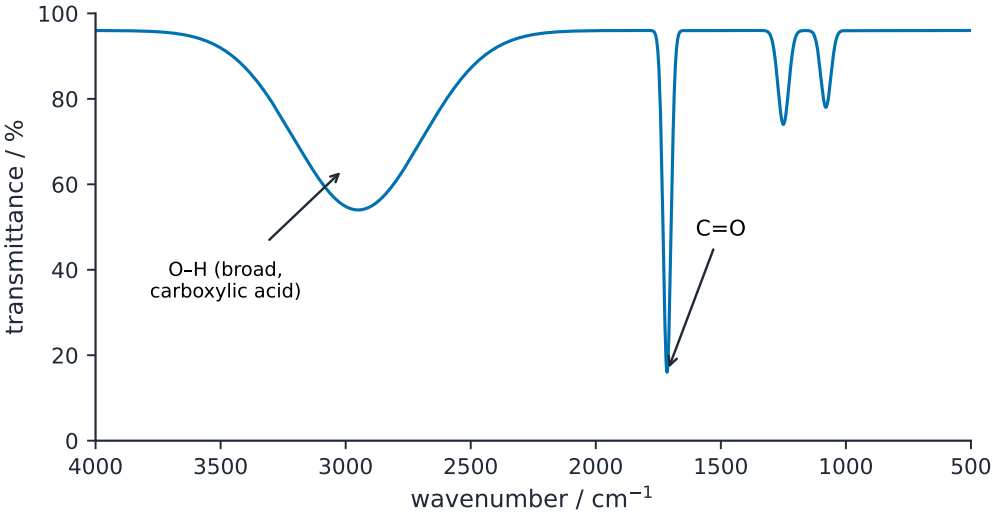


An infrared spectrometer shines infrared through a sample and records which wavenumbers its bonds absorb

The position is measured in **wavenumber** (in cm^{-1}). You are given a data table, so you do not memorise the numbers. You just match the dips to bonds:

- a broad dip around 3200–3650 cm^{-1} → an O–H bond in an alcohol.
- a dip around 1700 cm^{-1} → a C=O bond (aldehyde, ketone, acid or ester).
- a broad dip 2500–3000 cm^{-1} together with a C=O dip → a carboxylic acid.

This is useful for checking a reaction. For example, if propene has been turned into propan-2-ol, the C=C dip should be gone and an O–H dip should appear.



IR: dips at characteristic wavenumbers identify functional groups (e.g. C=O, O–H)

An infrared spectrum: each bond gives a dip at its own wavenumber. A broad O–H dip together with a C=O dip identifies a carboxylic acid

Mass spectrometry

In **mass spectrometry**, a molecule is turned into ions and sorted by its **mass-to-charge ratio** (m/e). The spectrum is a set of peaks at different m/e values.



A modern mass spectrometer: the sample is loaded at the front, then the machine ionises it and sorts the ions by their mass-to-charge ratio to give the spectrum

Relative atomic mass from isotopes

An element's **isotope** mixture gives several peaks. From the **isotopic abundance** (how common each isotope is) you can find the **relative atomic mass** – a weighted average:

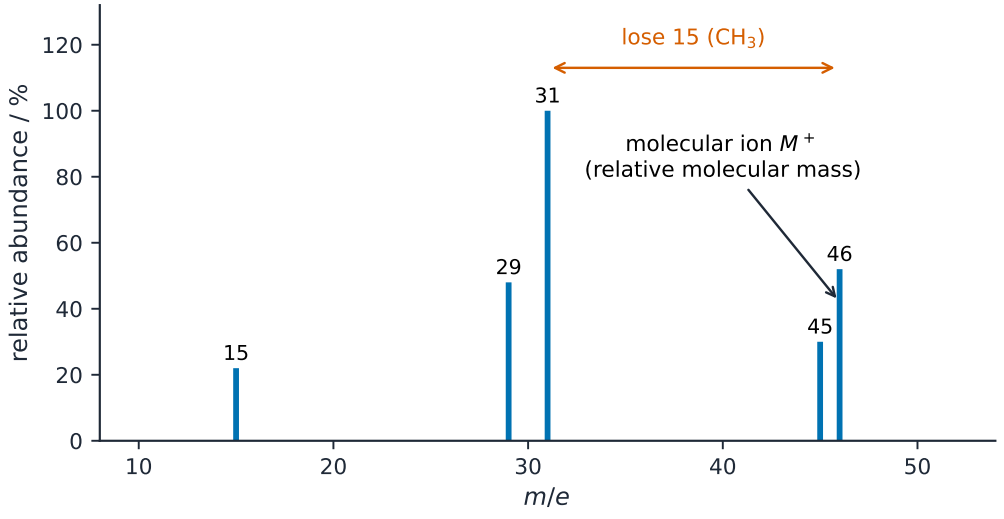
$$A_r = \frac{\sum(\text{isotope mass} \times \text{abundance})}{\sum \text{abundance}}$$

For example, chlorine is 75% ^{35}Cl and 25% ^{37}Cl, giving $A_r = \dfrac{35 \times 75 + 37 \times 25}{100} = 35.5$.

The molecular ion and fragmentation

The peak at the highest m/e (the **molecular ion** peak, or **molecular ion peak**, M^+) gives the relative molecular mass of the whole molecule.

The molecule also breaks into smaller pieces – this is **fragmentation**. The gap between two peaks tells you the mass of the lost piece, so you can suggest each **fragment**. For example, a loss of 15 means a CH_3 group was lost, and a loss of 29 means CHO or C_2H_5.



M^+ peak $\approx M_r$ · gaps between peaks = mass of fragments lost

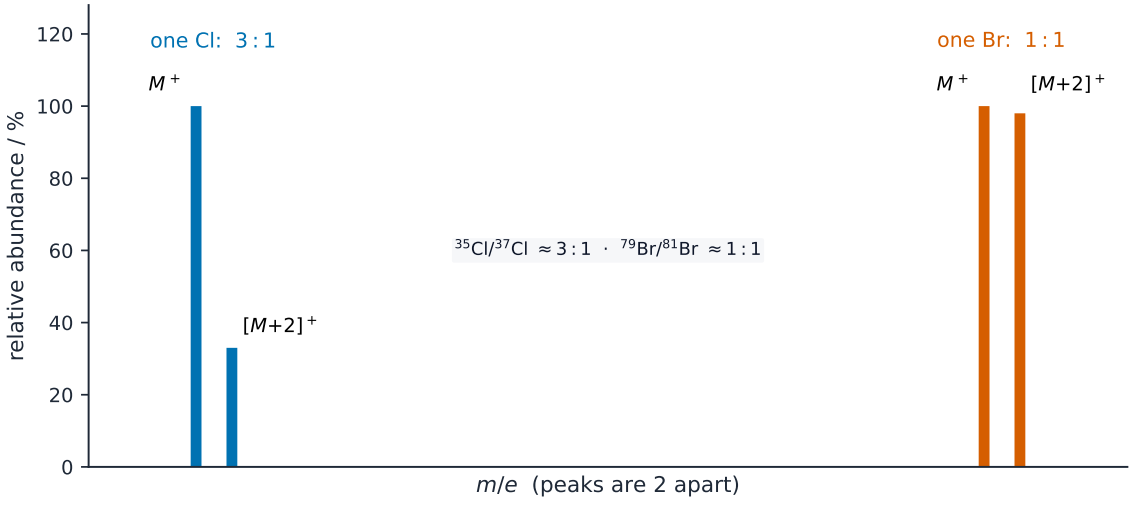
A mass spectrum: the highest-m/e peak is the molecular ion (M^+, the M_r); the gaps between peaks give the masses of the lost fragments

The [M + 1] and [M + 2] peaks

- a small **[M + 1]** peak comes from the ^{13}C isotope. The number of carbon atoms n is:

$$n = \frac{100 \times \text{abundance of } [M+1]^+}{1.1 \times \text{abundance of } M^+}$$

- an **[M + 2]** peak shows chlorine or bromine. One chlorine gives an $[M+2]$ peak about one third the height of M^+ (from ^{37}Cl); one bromine gives an $[M+2]$ peak about the **same** height as M^+ (from ^{81}Br).



An $[M+2]$ peak two mass units above M^+ shows a halogen: one chlorine gives a 3 : 1 ratio, one bromine a 1 : 1 ratio

Worked example. A compound shows a molecular ion at $m/e = 108$ and a peak of almost equal height at $m/e = 110$. Its infrared spectrum shows **no** broad absorption near 3300 cm^{-1}. Deduce its identity. Two peaks **two units apart with roughly equal heights** are the 1 : 1 signature of **one bromine** atom (from ^{79}Br and ^{81}Br); one chlorine would have given a 3 : 1 ratio instead. Take the bromine away from the molecular ion: $108 - 79 = 29$, which fits a C_2H_5 fragment. The absence of a broad peak near 3300 cm^{-1} rules out an O-H, so there is no alcohol group. The compound is **bromoethane**, C_2H_5Br. Read the $[M+2]$ **ratio** rather than merely noting the peak: 1 : 1 means bromine, 3 : 1 means chlorine.

Exam tips

- On an **IR spectrum** quote the **wavenumber and the bond** (O-H broad $\sim 3200 - 3600$, C=O ~ 1700); use the data booklet.
- The peak at the **highest** m/z is the molecular ion M^+ and gives the M_r.

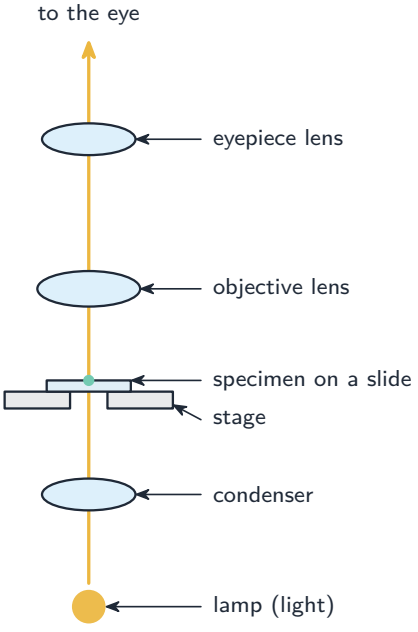
- Explain **fragmentation**: the gap between peaks is the lost fragment (loss of 15 = CH_3, 29 = C_2H_5 or CHO).
- Learn common **isotope patterns** (Cl gives M and M+2 in about 3:1; M+1 from ^{13}C).

Cell structure

A-Level Biology

How we study cells

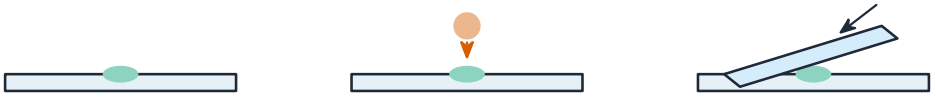
Cells are very small, so you cannot see them with your eyes alone. You use a **microscope** to make a bigger picture of them. The first kind you meet is the **light microscope**. It shines light through a thin **specimen** (the material you look at) and uses glass lenses to enlarge the view.



A light microscope: light passes up through the specimen, then two lenses (the objective and the eyepiece) magnify it

Making a slide and drawing what you see

To look at living material, you make a **temporary preparation**. You put a small, thin piece of material on a glass **slide**, add a drop of **stain** (a coloured liquid that makes parts easier to see), then lower a thin **cover slip** on top to flatten it and keep out air.



1. specimen on a slide
2. add a drop of stain
3. lower the cover slip at an angle

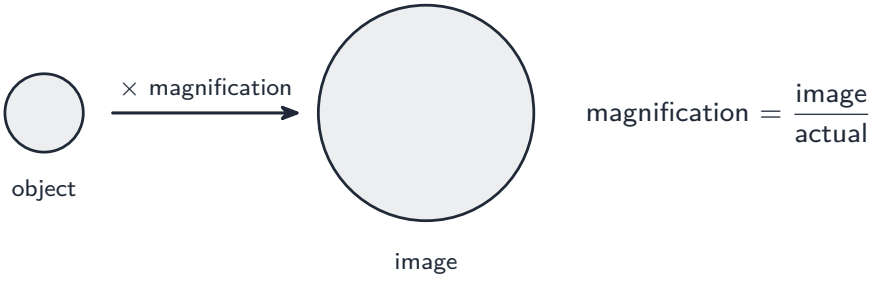
Making a temporary wet mount – lower the cover slip at an angle so no air bubbles are trapped

When you draw cells from a slide or a photograph, follow simple rules:

- use a sharp pencil and clear, single lines (no shading).
- draw only what you can really see, with the parts in the correct sizes.
- label the parts with straight lines that do not cross.

Magnification and actual size

Magnification tells you how many times bigger the image is than the real object. It has no unit. You find it with one equation:



Magnification = image size ÷ actual size

$$\text{magnification} = \frac{\text{size of image}}{\text{actual size of object}}$$

You can rearrange this to find any one value from the other two:

$$\text{actual size} = \frac{\text{size of image}}{\text{magnification}}$$

The top and the bottom of the fraction **must use the same unit**. Cells are tiny, so you work in small units:

- 1 mm = 1000 micrometre (µm)
- 1 µm = 1000 nanometre (nm)
- so 1 mm = 1 000 000 nm.

Worked example. In a photomicrograph at magnification 5000, a **chloroplast** measures 25 mm across. Its actual size is

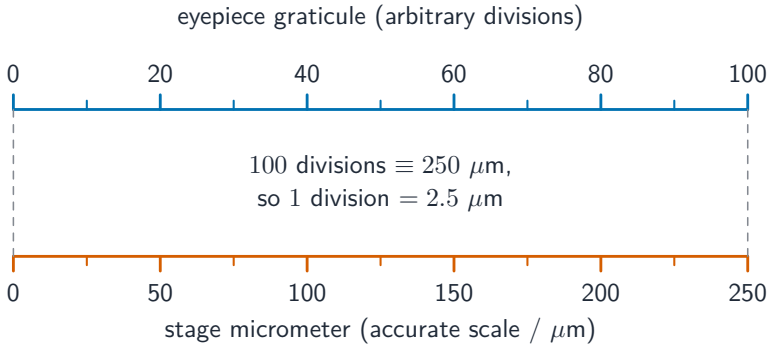
$$\frac{25 \text{ mm}}{5000} = 0.005 \text{ mm} = 5 \text{ µm}.$$

The same equation works for drawings, photomicrographs (photos taken through a light microscope) and **electron micrographs** (the most detailed photos, explained below). Always convert to the same unit first, then divide.

Eyepiece graticule and stage micrometer

To measure a real cell under the microscope, you use an **eyepiece graticule** – a tiny scale inside the eyepiece. Its divisions have no fixed size, so first you must **calibrate** them (work out what one division is worth).

You calibrate using a **stage micrometer** – a special slide with an accurate scale on it (often 1 mm split into 100 parts, so each part is 10 µm). You line up the two scales, count how many graticule divisions fit a known length, and divide. Once calibrated, you can swap in your specimen and measure it with the graticule.



Calibrate the graticule by lining it up with the stage micrometer's known scale

Resolution and magnification

These two words are easy to mix up. The examiner gives marks for the difference.

- **Magnification** is how many times bigger the image is than the object.
- **Resolution** is the smallest distance between two points that still lets you see them as **two** separate points.

Making an image bigger does not always show more detail. Past a certain point you just get a bigger, blurry image. Resolution sets the real limit on detail.

A light microscope has lower resolution because light has a fairly long **wavelength**. An **electron microscope** uses beams of **electrons** instead of light. Electrons have a much shorter wavelength, so the resolution is far higher and you can see very small structures inside the cell. There are two kinds: **scanning**

(shows the surface in 3D) and **transmission** (passes electrons through a thin slice to show inside detail).



light microscope:
the two blur into **one**

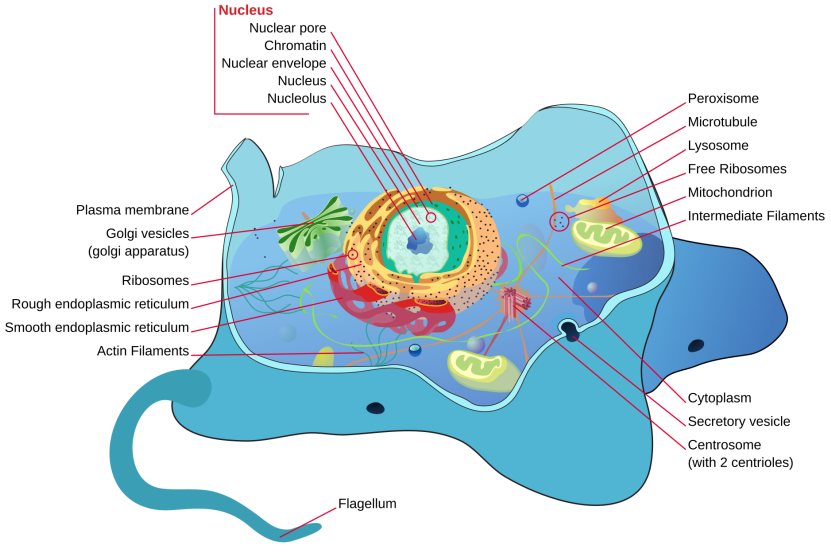


electron microscope:
resolved as **two**

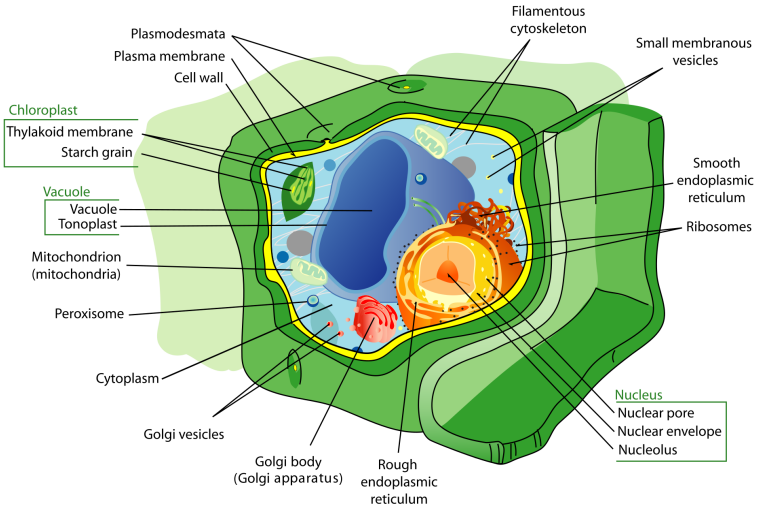
Resolution: a light microscope blurs two very close points into one; an electron microscope, with its shorter wavelength, resolves them as two

Eukaryotic cells and their organelles

Plant and animal cells are **eukaryotic cells** : their DNA is kept inside a **nucleus** . Inside the cell are many small parts called **organelles** , each with its own job. The jelly-like fluid around them is the **cytoplasm** .



A generalised animal cell and its organelles



A plant cell also has a cell wall, chloroplasts and a large vacuole

In a photomicrograph or electron micrograph you identify organelles by their shape, size and position; in a drawing you show their outlines and label them.

Organelle	Structure	Function
cell surface membrane	thin layer around the cell	controls what enters and leaves the cell
nucleus	large, surrounded by a **nuclear envelope** (a double membrane with holes); contains a **nucleolus**	holds the DNA; controls the cell; the nucleolus makes ribosomes
rough endoplasmic reticulum (rough ER)	sheets of membrane with ribosomes on the surface	makes and transports **proteins** (for example **antibodies**)
smooth endoplasmic reticulum (smooth ER)	sheets of membrane, no ribosomes	makes **lipids**

Organelle	Structure	Function
Golgi body	stack of flat membrane sacs	changes and packs proteins and lipids into **vesicles** for **secretion**
mitochondria	oval, with a folded inner membrane; has small circular DNA	site of **respiration** – releases energy as ATP
ribosomes	very small; 80S in the cytoplasm, 70S in chloroplasts and mitochondria	join **amino acids** to **synthesise** proteins
lysosomes	small sacs of **enzymes**	break down old organelles and waste
centrioles and **microtubules**	small tubes made of protein	help move chromosomes and form the cell's "skeleton"
cilia	tiny hairs on the cell surface that beat	move fluid or move the cell
microvilli	tiny folds of the cell surface membrane	increase surface area for **absorption**
chloroplasts (plants)	green, with stacked membranes; has small circular DNA	site of **photosynthesis**
cell wall (plants)	strong outer layer of **cellulose**	supports and protects the cell; stops it bursting
plasmodesmata (plants)	tiny channels through the cell walls	link the cytoplasm of neighbouring cells
large permanent **vacuole** (plants)	big sac of watery fluid, with a membrane called the **tonoplast**	stores water and keeps the cell firm

Cells use **ATP** made in respiration as their energy supply for every job that needs energy, such as making proteins, moving things and dividing.

Comparing plant and animal cells

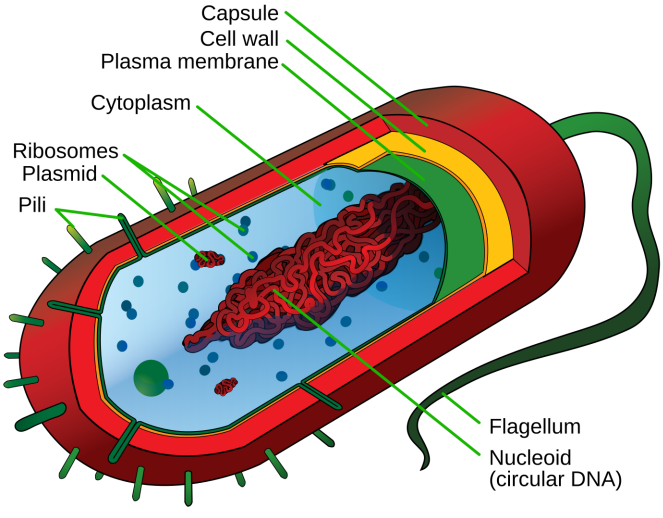
Feature	Plant cell	Animal cell
cell wall	present (cellulose)	absent
chloroplasts	present	absent
large permanent vacuole	present	absent (only small, temporary ones)
centrioles	absent in most	present
shape	fixed and regular	rounder and more flexible

Both have a cell surface membrane, cytoplasm, a nucleus, mitochondria, ribosomes, ER and a Golgi body.

Prokaryotic cells (bacteria)

A **prokaryotic cell** , such as a **bacterium** , is much smaller and simpler than a eukaryotic cell. Its key features are:

- **unicellular** – it is a single cell.
- generally 1–5 µm across.
- a cell wall made of **peptidoglycan** (not cellulose).
- circular DNA lying free in the cytoplasm – there is **no nucleus**.
- 70S ribosomes (smaller than the 80S ones in the cytoplasm of eukaryotes).
- **no organelles surrounded by a double membrane** – so no nucleus, no mitochondria and no chloroplasts.



A prokaryotic cell: the circular DNA (nucleoid) lies free, with no nucleus and no double-membrane organelles

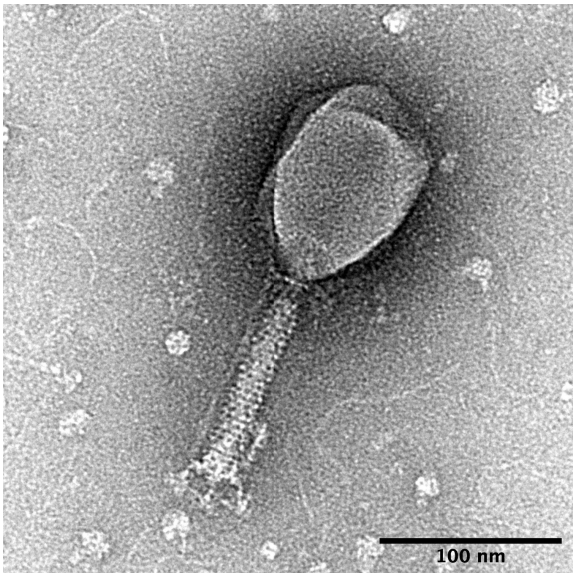
■ **Comparing prokaryotic and eukaryotic cells**

Feature	Prokaryotic cell	Eukaryotic cell
size	about 1–5 µm	about 10–100 µm
DNA	circular, free in cytoplasm	linear, inside a nucleus
nucleus	none	present
double-membrane organelles	none	mitochondria (and chloroplasts in plants)
ribosomes	70S	80S (with 70S inside mitochondria and chloroplasts)
cell wall	peptidoglycan	cellulose (plants) or none (animals)

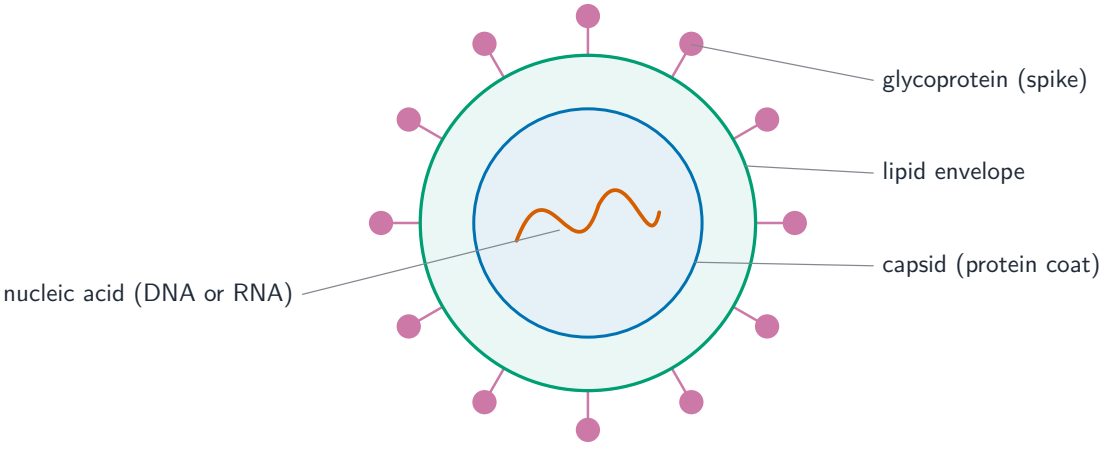
Viruses

All **viruses** are **non-cellular** – they are not made of cells at all. Each virus is built from just two or three parts:

- a core of **nucleic acid**, which is **either DNA or RNA** (never both).
- a protein coat around the core called a **capsid**.
- in some viruses, an outer **envelope** made of **phospholipids**.



A real virus, magnified hugely. The rounded head is the protein capsid wrapped around the nucleic acid core; the tail injects that nucleic acid into a bacterium. Note the scale bar – the whole virus is about 100 nm, far smaller than any cell



A generalised virus: nucleic acid inside a protein capsid, with a lipid envelope in some viruses

A virus has no cytoplasm, no organelles and no ribosomes. It cannot respire or make its own proteins. It can only copy itself inside a living **host** cell, so it sits at the edge of what we call "living".

Exam tips

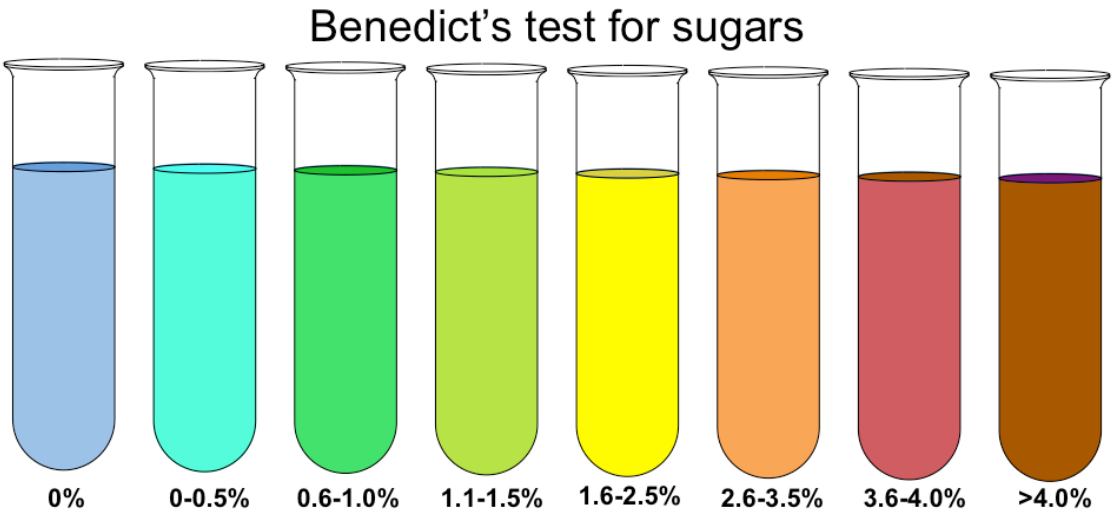
- Magnification = image size ÷ actual size – **convert units first** (mm ↔ μm ↔ nm), then rearrange for whichever is unknown.
- Distinguish **magnification** (how many times larger) from **resolution** (smallest distance still seen as two points); electron microscopes resolve more because electrons have a shorter wavelength.
- Give each organelle a **structure + function** pair (e.g. mitochondrion: folded inner membrane → aerobic respiration).
- State the **prokaryote vs eukaryote** differences exactly: no nucleus, smaller (70S) ribosomes, no membrane-bound organelles, circular DNA.
- Viruses are **non-living** – describe them only by capsid, genetic material and (sometimes) an envelope.

Biological molecules

A-Level Biology

Testing for biological molecules

Living things are built from four main kinds of biological **molecule** : **carbohydrates** , **lipids** , **proteins** and **nucleic acids** . You can use simple chemical tests to find out which kinds are present in a sample.



Benedict's test: blue turns green, then orange, then brick-red as more reducing sugar is present

Test	What it finds	Method	Positive result
Benedict's test	**reducing sugar**	add Benedict's solution and heat in a water bath	a green, yellow, orange or brick-red **precipitate**
iodine test	**starch**	add orange-brown **iodine** solution	colour changes to blue-black
emulsion test	**lipid**	mix sample with ethanol, then pour into water	a white, cloudy **emulsion** forms
biuret test	**protein**	add biuret solution at room temperature	blue changes to purple

Semi-quantitative Benedict's test

The normal Benedict's test only tells you "yes or no". A **semi-quantitative** test gives a rough amount. First you **standardise** the test: you run it on solutions of known **concentration** and record the result for each. Then you can estimate an unknown by either:

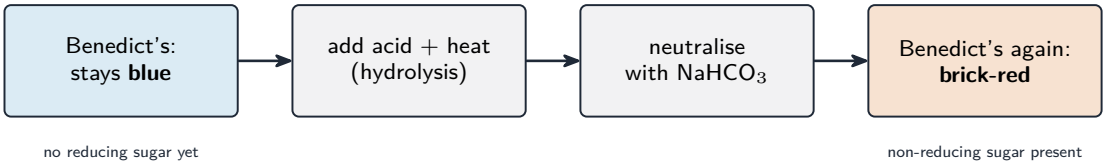
- the **time to the first colour change** (more sugar changes colour faster), or
- comparing the final colour to your set of colour standards.

Keep sample and reagent volumes, water-bath temperature and the colour-change endpoint the same for standards and the unknown. Use a water blank, repeat readings and use closer standards around the estimate; do not assume the time is proportional to concentration.

Testing for non-reducing sugars

Some sugars, such as sucrose, are **non-reducing sugar** : they give a negative Benedict's test. To detect them:

1. Do a normal Benedict's test first. It stays blue (reducing sugar is not detected under these conditions).
2. Take a fresh sample and add dilute **hydrochloric acid** , then heat. This **acid hydrolysis** breaks the sugar into smaller reducing sugars.
3. Cool, then **neutralise** the acid with sodium hydrogencarbonate.
4. Add Benedict's solution and heat in a water bath. A coloured precipitate after an initially negative test supports a non-reducing sugar in a sample known to contain sugars; it does not identify sucrose specifically.



Testing for a non-reducing sugar: hydrolyse it with acid first, then the second Benedict's test turns red

Worked example. A solution is known to contain one dissolved sugar and no starch. One portion gives a negative Benedict's test. A fresh portion is heated with dilute hydrochloric acid, cooled, neutralised and re-tested with Benedict's in a water bath; it gives an orange precipitate. This supports a non-reducing sugar: hydrolysis releases reducing monosaccharides, and neutralisation restores the alkaline conditions needed for the test. The sugar is not identified as sucrose by this result alone. Without the sample context, other carbohydrates that release reducing sugars on hydrolysis must also be considered. A positive initial test followed by another positive test alone would not establish an additional non-reducing sugar.

Carbohydrates

■ Monomers, polymers and macromolecules

- a **monomer** is a small molecule that is a single unit.
- a **polymer** is a long molecule made of many monomers joined together.
- a **macromolecule** is any very large molecule.

Sugars come in three sizes:

- a **monosaccharide** is a single sugar unit, such as **glucose** and fructose.
- a **disaccharide** is two units joined, such as maltose and sucrose.
- a **polysaccharide** is many units joined into a polymer.

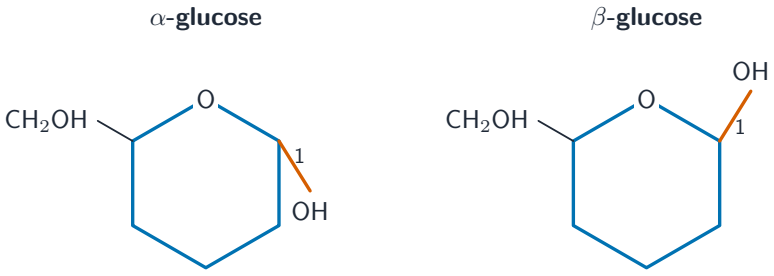
Glucose, **fructose** and **maltose** are reducing sugars. **Sucrose** is a non-reducing sugar.

■ Two ring forms of glucose

Glucose has six carbon atoms and forms a ring. There are two ring forms. They differ only at carbon 1:

- in α**-glucose**, the –OH group on carbon 1 points **down**, below the ring.
- in β**-glucose**, the –OH group on carbon 1 points **up**, above the ring.

This small difference decides which polysaccharide the glucose can build.



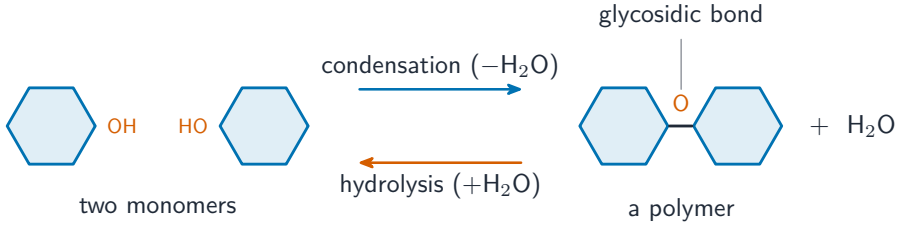
only carbon 1 differs: the OH points **down** in α, **up** in β

The two ring forms differ only at carbon 1: the –OH points down in α, up in β

Joining and breaking sugars

Monomers are joined by strong **covalent bonds** . When two sugars join, a **glycosidic bond** forms between them. This happens by **condensation** : a molecule of water is removed each time a bond forms.

The reverse is **hydrolysis** : a water molecule is added to break a glycosidic bond. This is why the non-reducing sugar test needs acid and heat – they hydrolyse sucrose into glucose and fructose.



Condensation removes water to join monomers; hydrolysis adds water to split them

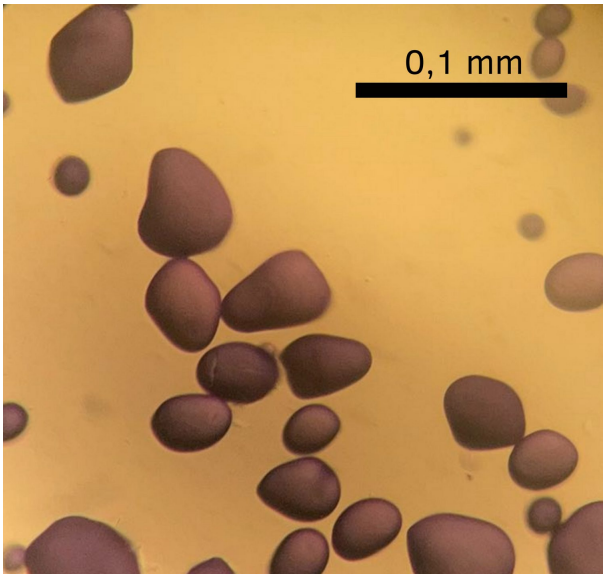
Storage polysaccharides: starch and glycogen

Starch is the energy store in plants. It is made of two polymers of α-glucose:

- **amylose** – a long, unbranched chain that coils into a spiral.
- **amylopectin** – a chain with many side branches.

Glycogen is the energy store in animals. It is like amylopectin but has even more branches, so it can be broken down quickly when energy is needed.

These stores suit their job well: they are compact, they are **insoluble** (so they do not leave the cell), and they do not change the **water potential** of the cell (so they do not pull water in by **osmosis**). The many branches give many ends, so glucose can be added or removed fast.



Real starch grains inside a potato, stained by iodine. Each grain is a dense, insoluble package of amylose and amylopectin – note the scale: the largest are only about 0.1 mm across

Cellulose

Cellulose is made of β-glucose. Because of the β form, every other glucose is flipped over, so the chains are long and straight. Many straight chains lie side by side and are held together by **hydrogen bonds** into strong bundles called **microfibrils**. These give the plant **cell wall** its strength and stop the cell bursting.



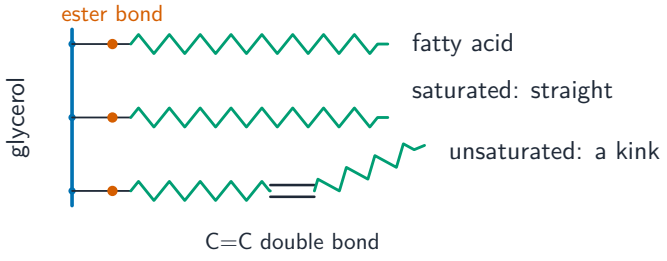
The shape fits the job: amylose coils and amylopectin/glycogen branch (for compact stores), while straight cellulose chains pack into strong fibres

Lipids

Triglycerides

A **triglyceride** is the main fat or oil. It is **non-polar** and **hydrophobic** (it does not mix with water). It is made from one **glycerol** molecule joined to three **fatty acids** by **ester bonds**. Each ester bond

forms by condensation, so three water molecules are removed.



One glycerol plus three fatty acid tails; a straight tail is saturated, a kinked one unsaturated

Fatty acids are of two kinds:

- **saturated** – no carbon–carbon **double bonds** ; these fats are usually solid.
- **unsaturated** – one or more double bonds; these oils are usually liquid.

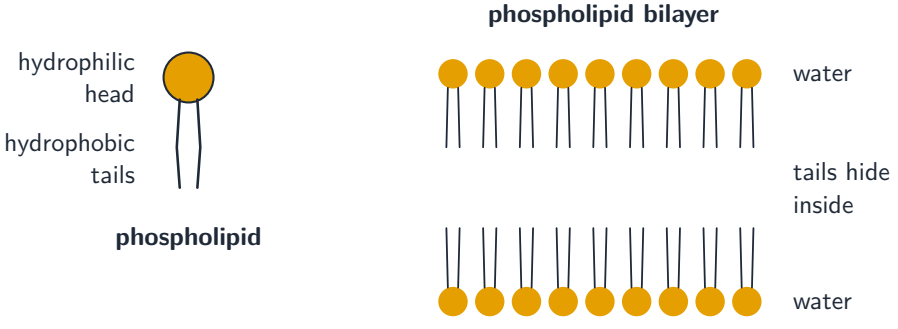
Triglycerides make a good long-term energy store: they release about twice as much energy per gram as carbohydrates, they are insoluble, and they store little extra mass because they hold no water. Under the skin they also give **insulation** and protect the organs.

Phospholipids

A **phospholipid** is like a triglyceride, but one fatty acid is replaced by a **phosphate** group. This gives the molecule two ends with different behaviour:

- a **hydrophilic** ("water-loving") **polar** phosphate head.
- two hydrophobic ("water-fearing") fatty acid tails.

This split personality is why phospholipids form the membranes around cells.



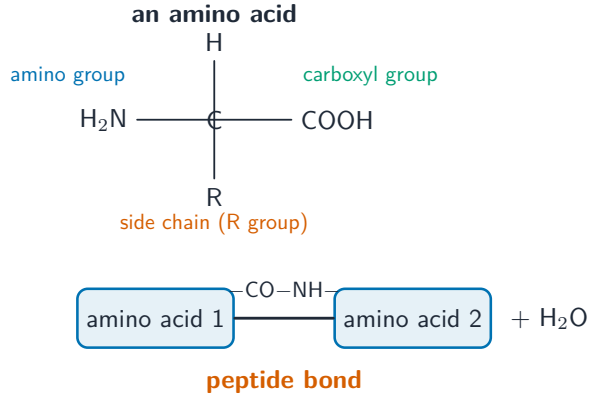
The hydrophilic heads face the water; the hydrophobic tails hide inside, forming a bilayer

Proteins

Amino acids and the peptide bond

Proteins are polymers of **amino acids** . Every amino acid has the same general structure around a central carbon atom: an **amino group** ($–NH_2$), a **carboxyl group** (–COOH), a hydrogen atom, and a variable **side chain** (the R group). The R group is different in each amino acid.

Two amino acids join by condensation. The bond formed between the amino group of one and the carboxyl group of the next is a **peptide bond** , and a water molecule is removed. Many amino acids joined this way make a **polypeptide** . Adding water (hydrolysis) breaks a peptide bond.

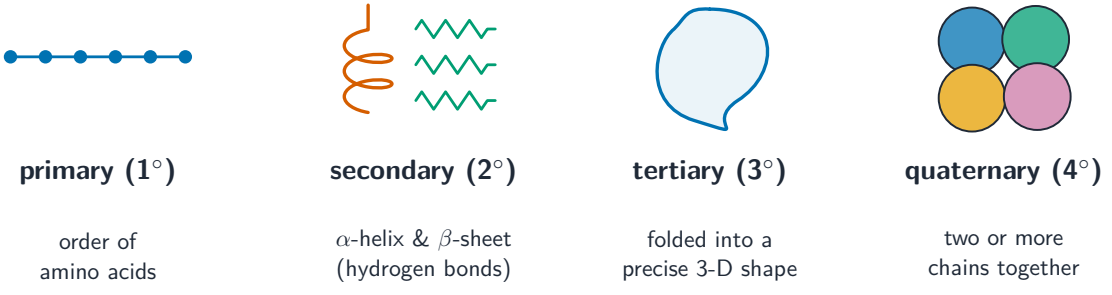


condensation removes one water for each peptide bond

Every amino acid has an amino group, a carboxyl group and an R group; two join by a peptide bond

Four levels of protein structure

Level	What it means
primary structure	the order of amino acids in the chain
secondary structure	local shapes – the **α-helix** and the **β-pleated sheet** – held by hydrogen bonds between backbone groups
tertiary structure	the whole chain folded into a precise 3-D shape
quaternary structure	two or more polypeptide chains joined into one protein



The four levels: primary → secondary → tertiary → quaternary structure

The tertiary folding is stabilised by several kinds of interaction involving R groups:

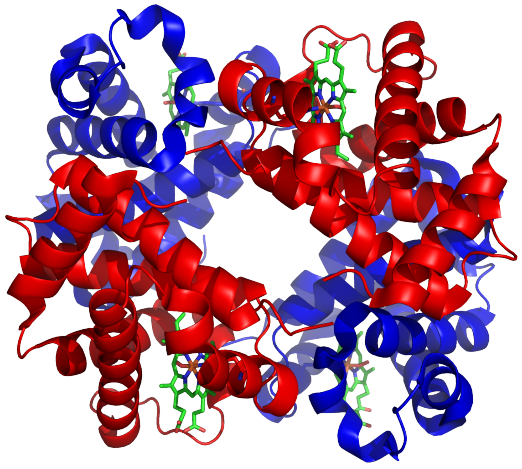
- **hydrophobic interactions** (non-polar R groups cluster away from water).
- hydrogen bonding.
- **ionic bonds** (between charged R groups).
- covalent bonding, including strong **disulfide bonds** .

Globular and fibrous proteins

- **globular proteins** fold into a rounded shape, are usually **soluble** , and do jobs in the body (for example enzymes and haemoglobin).
- **fibrous proteins** form long strands, are usually insoluble, and give structure and support (for example collagen).

Haemoglobin – a globular protein

Haemoglobin carries **oxygen** in red blood cells. It has a quaternary structure made of four polypeptide chains: two alpha (α-globin) chains and two beta (β-globin) chains. Each chain holds a **haem group** . At the centre of each haem group is an **iron** ion, and this is where one oxygen molecule binds. Four chains mean one haemoglobin molecule can carry four oxygen molecules.



A real haemoglobin molecule, worked out from X-ray data. Count them: two alpha chains (red), two beta chains (blue), and one green haem group held in each – so four oxygen molecules in total

Collagen – a fibrous protein

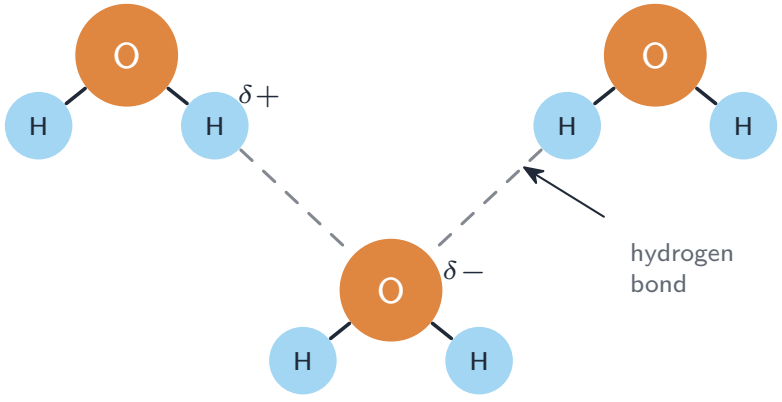
Collagen gives strength to skin, **tendons** , bone and blood vessel walls. One collagen molecule is three helical polypeptide chains wound tightly into a triple helix, stabilised by hydrogen bonds between chains. Glycine occurs at about every third position; its small side chain allows close packing. Many of these molecules lie side by side, slightly staggered, and have covalent cross-links between molecules, forming thick **fibres** . The staggered, cross-linked arrangement makes collagen very strong when pulled.

Water

Water is a small molecule, but its two O–H bonds are **polar**: the oxygen end is slightly negative and the hydrogen ends are slightly positive. So one water molecule attracts its neighbours, forming weak hydrogen bonds between them. These hydrogen bonds explain water's useful properties:

- **solvent action** – water is a good **solvent** , so many substances dissolve in it. This lets reactions happen and lets substances be carried around the body.

- **high specific heat capacity** – water needs a lot of energy to warm up, so its temperature stays steady. This protects living things from quick temperature changes.
- **latent heat of vaporisation** – water needs a lot of energy to **evaporate** . So when water evaporates (for example as sweat dries), it carries away a lot of heat and cools the body.



Water is polar (δ− oxygen, δ+ hydrogens), so its molecules attract each other by hydrogen bonds – the reason for all the properties above

Exam tips

- Learn each **food test** as reagent + positive result + colour change (Benedict's → coloured precipitate, not necessarily brick-red; iodine → blue-black; biuret → purple; emulsion → white).
- Benedict's detects **reducing** sugars; for a non-reducing sugar you must heat a fresh portion with acid, cool and neutralise it, then re-test in a water bath.
- Name the bond precisely: **glycosidic** (carbohydrates), **ester** (lipids), **peptide** (proteins) – all made by **condensation** (water removed).
- Link structure to function: cellulose (straight chains, H-bonds → strong), glycogen/starch (branched/coiled → compact store).
- For proteins, name the bond at each level: primary (peptide), secondary (hydrogen), tertiary (R-group interactions), quaternary.

Enzymes

A-Level Biology

What enzymes are

Enzymes are **globular proteins** – a type of **protein** with a rounded, soluble shape. They are biological catalysts: they **catalyse** (speed up) the chemical reactions in living things, and they are not used up, so each enzyme works again and again.

Enzymes work in two places:

- **intracellular** enzymes work inside the **cell** that made them. An example is **catalase**, which breaks down harmful hydrogen peroxide.
- **extracellular** enzymes are **secreted** (sent out) to work outside the cell. An example is **amylase**, which is released into the gut to **digest** **starch**.



Yeast enzymes ferment sugar, giving off bubbles of carbon dioxide

How enzymes work

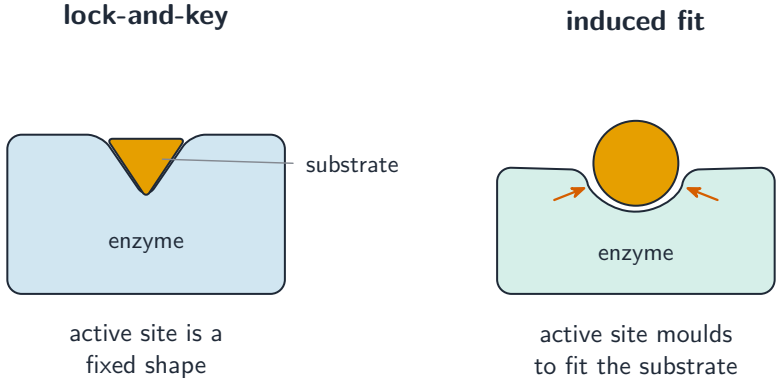
Each enzyme has a special pocket called the **active site**. The molecule it acts on is its **substrate**. The substrate fits into the active site to form an **enzyme–substrate complex**. The reaction then happens, and the **products** leave, freeing the active site for the next substrate.

Specificity

An enzyme is **specific**: it usually works on only one substrate. This is because the shape of the active site is **complementary** to (fits) the shape of that substrate and no other. We call this **specificity** .

Two ideas explain how the substrate fits:

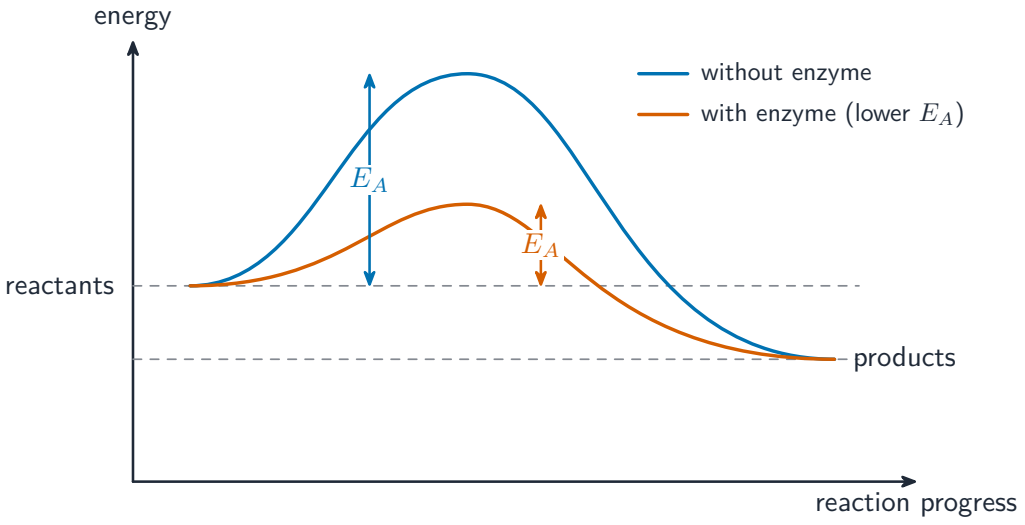
- the **lock-and-key hypothesis** – the active site is a fixed shape, and only a substrate with the matching shape fits, like a key in a lock.
- the **induced-fit hypothesis** – the active site is not quite the right shape at first. When the substrate binds, the active site changes shape a little to wrap around it tightly. This idea fits the evidence better.



Lock-and-key: a fixed active site. Induced fit: the active site changes shape to grip the substrate

Lowering activation energy

Every reaction needs a small "push" of energy to start, called the **activation energy** . An enzyme lowers the activation energy. This lets the reaction go quickly at the cell's normal **temperature** , instead of needing high heat.



The enzyme route has a lower activation energy (E_A), so more molecules can react

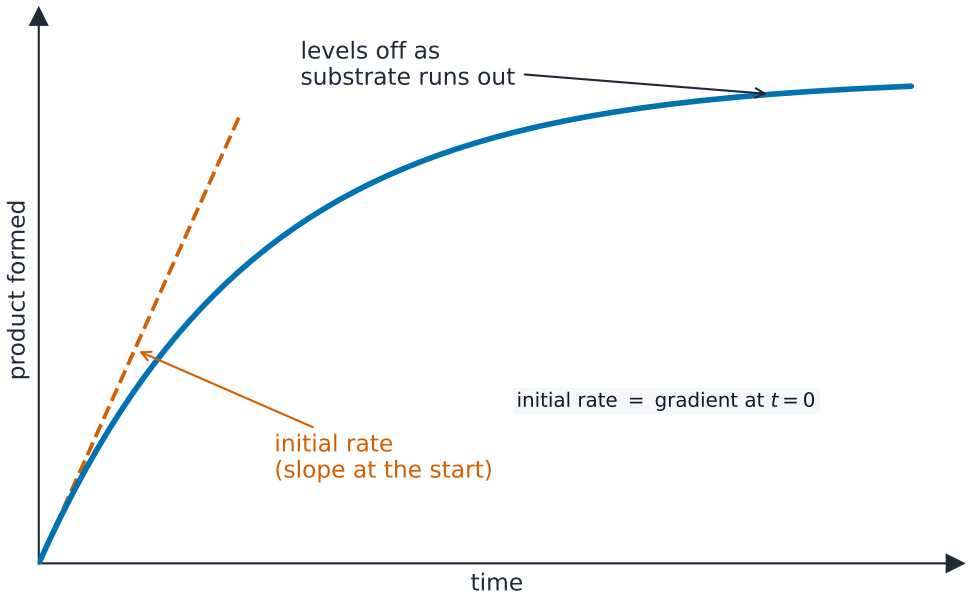
Measuring the rate of a reaction

You can follow an enzyme reaction in two ways:

- measure how fast **product** is made. With catalase, oxygen gas is a product, so you collect the gas and measure its volume over time.
- measure how fast substrate disappears. With amylase, you remove samples and use the iodine test; the blue-black colour fades as the starch is used up.

A **colorimeter** makes this exact. It shines light through the tube and measures how much light is absorbed, so a colour change becomes a number you can plot.

The **rate of reaction** is steepest at the start (most substrate present), so the **initial rate** (the slope at time zero) is the fairest value to compare.



Product builds fastest at the start; the initial rate is the slope of the tangent at time zero – the fairest value to compare

Worked example. In the first 20 seconds of a catalase reaction, 16 cm^3 of oxygen is collected. Estimate the rate of reaction.

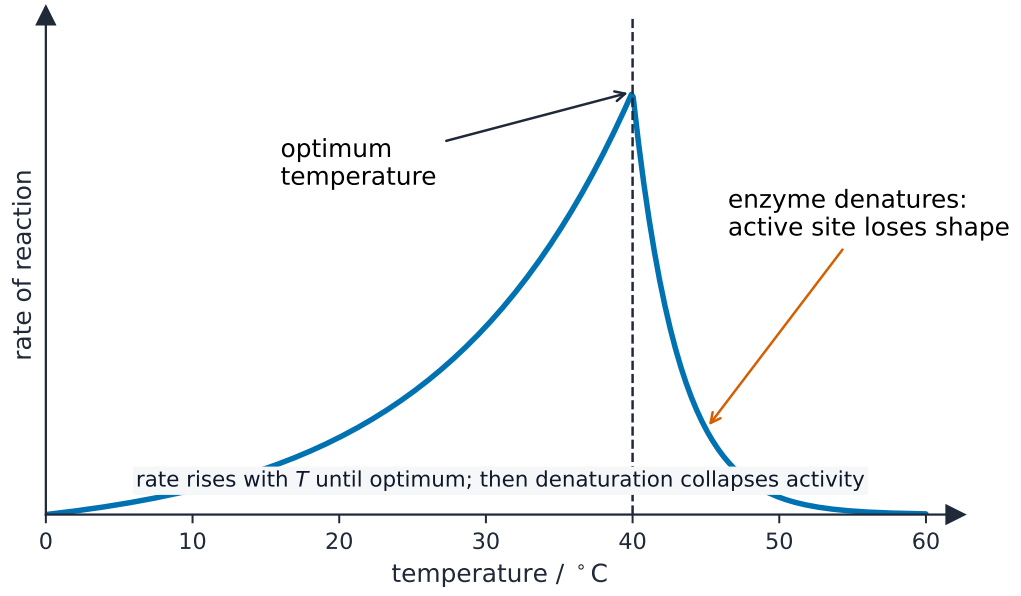
$$\text{rate} = \frac{\text{volume of product}}{\text{time}} = \frac{16}{20} = 0.8 \text{ cm}^3\,\text{s}^{-1}.$$

Because the reaction is fastest at the start, measuring over this short early interval gives a value close to the initial rate; averaging over a longer time would include the slower later stages and underestimate it.

Factors that affect enzyme activity

Temperature

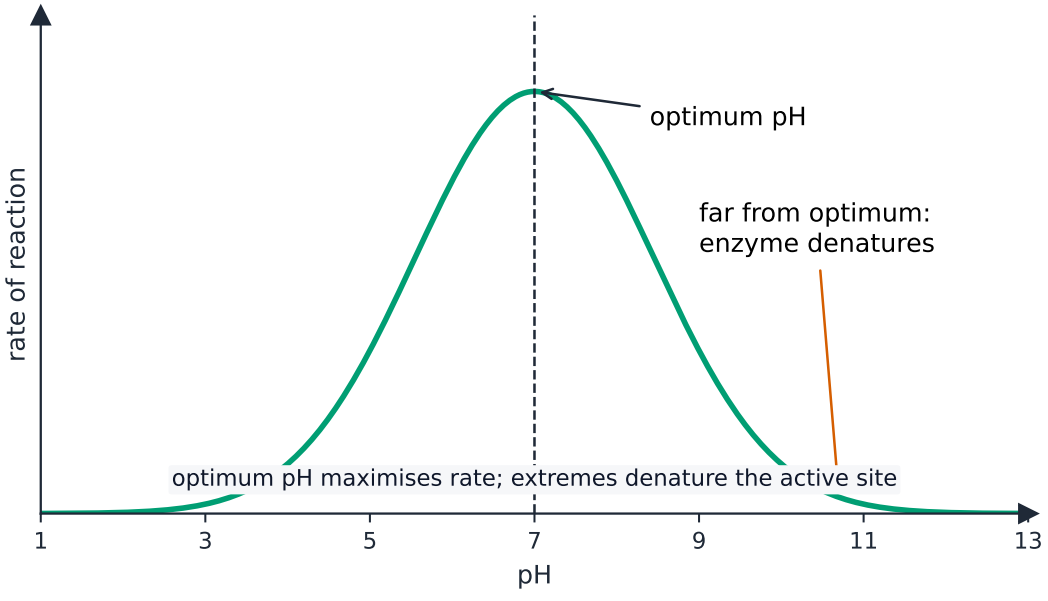
As temperature rises, molecules gain more **kinetic energy** and **collide** more often, so the rate rises. But above the **optimum temperature** the enzyme begins to **denature**: the heat breaks the bonds holding its shape, so the active site changes and no longer fits the substrate. The rate then falls quickly.



Rate rises to the optimum, then falls fast as the enzyme denatures

■ pH

Each enzyme has an optimum pH. If the pH moves too far from it, the enzyme denatures and the rate drops. To study pH fairly, you keep it steady with a **buffer solution** .



The rate peaks at the optimum pH and falls away on either side

Enzyme concentration

With plenty of substrate, more enzyme means more active sites, so the rate goes up in proportion to enzyme concentration.

Substrate concentration

At first, adding more substrate speeds the reaction. But once every active site is busy, adding more makes no difference – the rate levels off at a maximum.

Inhibitor concentration

An **inhibitor** is a molecule that slows an enzyme. The more inhibitor present, the lower the rate.

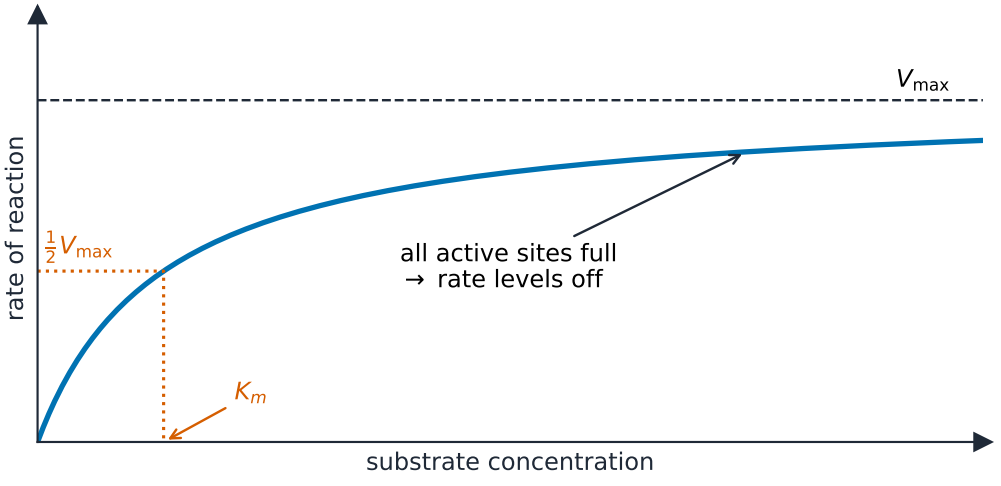
V_max and the Michaelis–Menten constant

The levelling-off rate, when all active sites are full, is the maximum rate, written $V_{\max}$.

The **Michaelis–Menten constant** (K_{m}) is the substrate concentration that gives **half** of $V_{\max}$. It tells you about the enzyme's **affinity** (pulling power) for its substrate:

- a **low** K_{m} means the enzyme reaches half-speed at a low substrate concentration, so it has a **high** affinity.
- a **high** K_{m} means a **low** affinity.

So K_{m} lets you compare how strongly different enzymes hold their substrates.



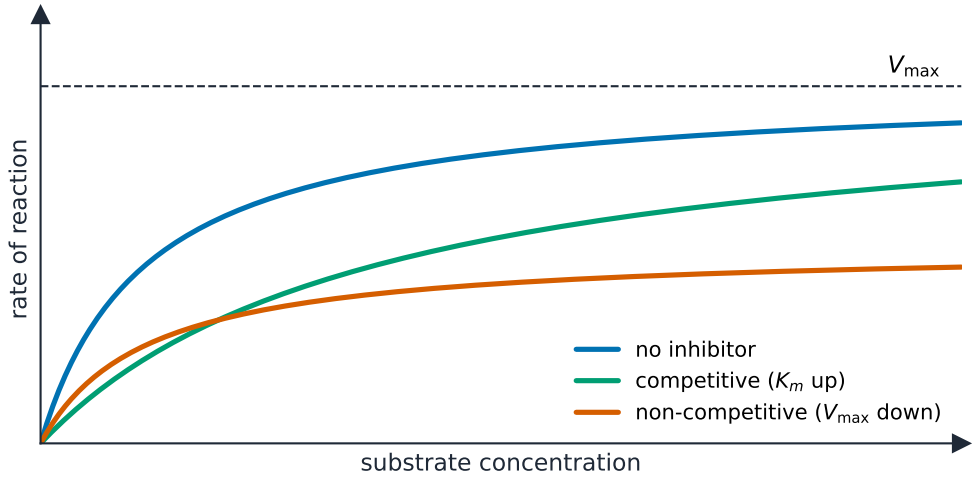
$v = \frac{V_{\max}[S]}{K_m + [S]}$ · K_m = [S] at half $V_{\max}$

Rate climbs to V_{max} when all active sites are full; K_m is the substrate concentration giving half V_{max}

Reversible inhibitors

Some inhibitors are **reversible** : they can leave the enzyme again. There are two types.

Type	Where it binds	Effect of adding more substrate	Effect on V_{max} and K_m
competitive inhibitor	in the active site (it has a similar shape to the substrate)	more substrate out-competes it, so its effect is reduced	V_{max} unchanged; K_m rises
non-competitive inhibitor	at another site, changing the active site's shape	adding more substrate does not help	V_{max} falls; K_m unchanged

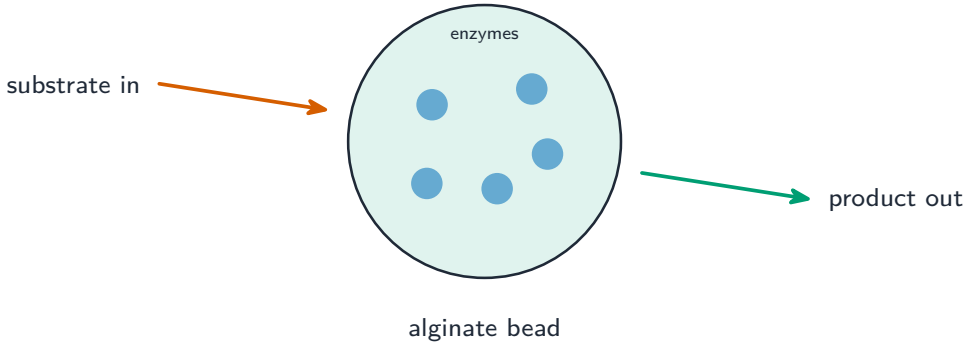


competitive: ↑ K_m, same V_{max} · non-competitive: ↓ V_{max}

A competitive inhibitor raises K_m (more substrate overcomes it); a non-competitive one lowers V_{max}

Immobilised enzymes

An **immobilised enzyme** is fixed in place – for example, trapped inside small beads of **alginate** – instead of floating free in solution. The substrate solution flows past the beads.



The enzymes stay trapped in the bead while substrate flows in and product flows out – so the enzyme is never washed away

A free enzyme usually works a little faster, because the substrate can reach it easily. But immobilised enzymes have big practical advantages:

- the enzyme is not washed away, so it can be used again and again.
- the product is pure – it is not mixed with enzyme.
- the enzyme is more stable, so it survives changes in temperature and pH better.
- the process can run continuously, with substrate flowing in and product flowing out.



Biological washing powders contain enzymes that digest food and blood stains at low temperatures

Exam tips

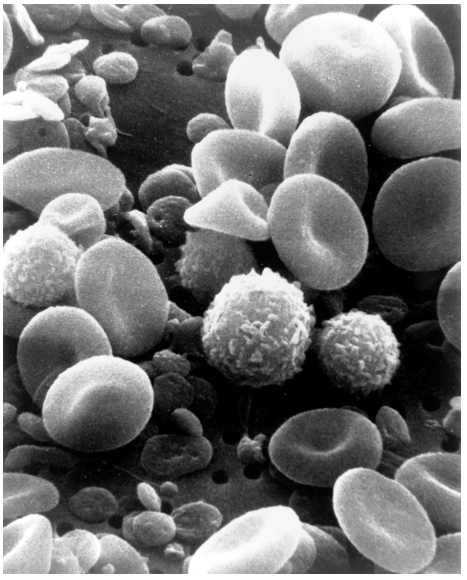
- Explain enzyme action with the **induced-fit** model (the active site moulds around the substrate) – now preferred to lock-and-key.
- Above the optimum the enzyme **denatures** (hydrogen bonds and tertiary structure break) – write "denatures", never "dies" or "is killed".
- Distinguish inhibitors: **competitive** binds the active site (overcome by more substrate, same V_{max}); **non-competitive** binds elsewhere (lower V_{max}).
- Compare rates using the **initial rate** (tangent at time zero) – the fairest measure, before substrate becomes limiting.

Cell membranes and transport

A-Level Biology

The cell surface membrane

Every cell is wrapped in a **cell surface membrane** . We describe its structure with the **fluid mosaic model** .



Red blood cells under a scanning electron microscope – each is wrapped in a cell surface membrane

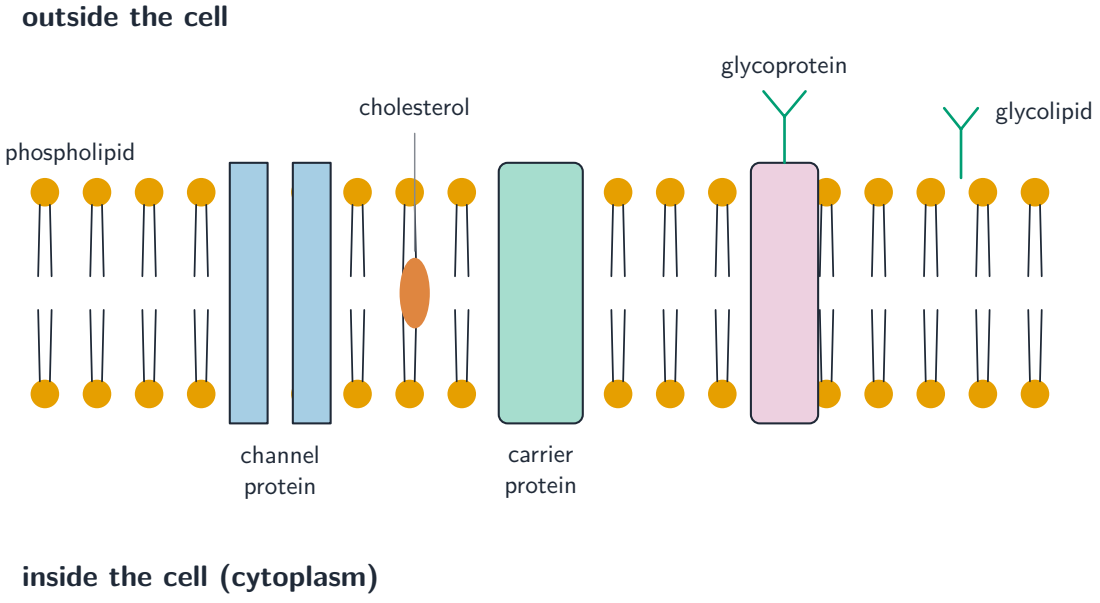
The phospholipid bilayer

The membrane is built mainly from **phospholipids** . Each phospholipid has a **hydrophilic** ("water-loving") head and two **hydrophobic** ("water-fearing") tails. There is water on both sides of the membrane, so the phospholipids line up in two layers – a **bilayer** – with the heads facing the water outside and inside, and the tails hidden in the middle, away from water. This arrangement forms by itself because of those hydrophilic and hydrophobic interactions.

The model is called "fluid" because the phospholipids are not fixed: they slide past each other, so the membrane can move and bend. It is called a "mosaic" because many **proteins** are dotted through it, like tiles in a picture.

What floats in the membrane

Part	Where it sits	Main roles
phospholipids	the two layers	form the basic barrier
proteins	through the membrane or on its surface	transport, support and signalling
carrier proteins	span the membrane	carry specific molecules across
channel proteins	span the membrane	form water-filled pores for ions to pass
cholesterol	between the phospholipid tails	controls **fluidity** and adds strength
glycolipids and **glycoproteins**	carbohydrate chains on the outer surface	cell recognition; some act as **antigens**



The fluid mosaic model: proteins, cholesterol and carbohydrate chains sit in a fluid phospholipid bilayer

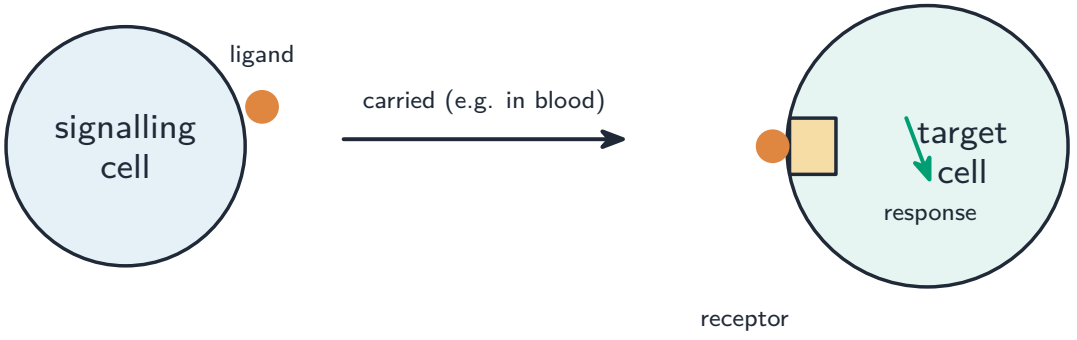
So the membrane molecules together give the membrane its **stability**, its fluidity, its **permeability** (control over what gets through), its transport jobs, its signalling jobs, and its cell recognition.

The membrane is **partially permeable**: it lets some substances through easily but blocks others.

Cell signalling

Cells talk to each other by **cell signalling** . The main stages are:

1. a cell **secretes** a signal chemical called a **ligand** (for example a **hormone**).
2. the ligand is carried (often in the blood) to a **target cell** .
3. the ligand binds to a specific **receptor** on the target cell's surface membrane. The shape of the receptor matches that ligand only. Binding then triggers a particular response inside the target cell.



The ligand fits one receptor shape only, so only cells with that receptor respond to the signal

Moving substances across the membrane

There are six processes. Some are **passive** (they need no energy), and some are **active** (they use energy from ATP).

Simple diffusion

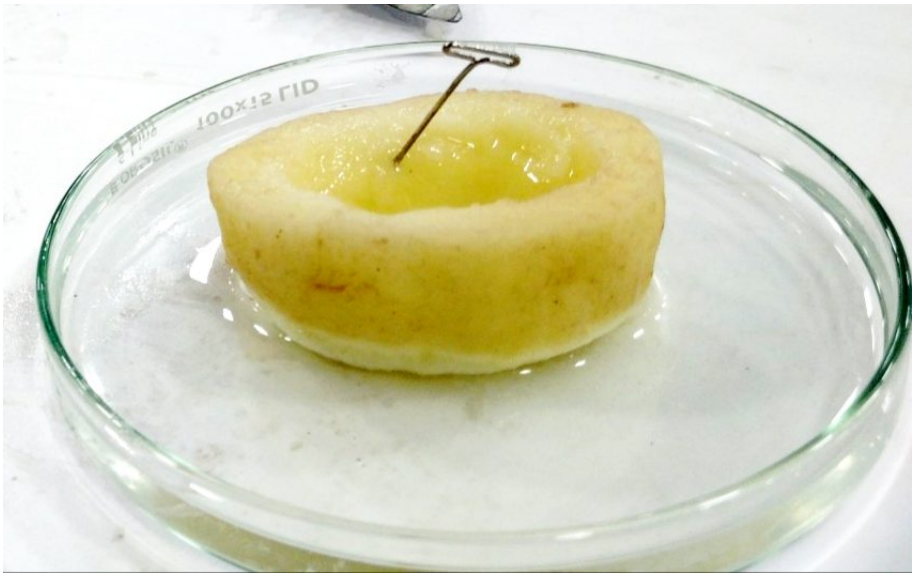
Diffusion is the net movement of particles from where they are at a high **concentration** to where they are at a low concentration, until they are spread evenly. **Simple diffusion** is when particles pass straight through the bilayer, down the **concentration gradient** . Only small or non-polar molecules can do this – such as **oxygen** and **carbon dioxide** . It is passive.

Facilitated diffusion

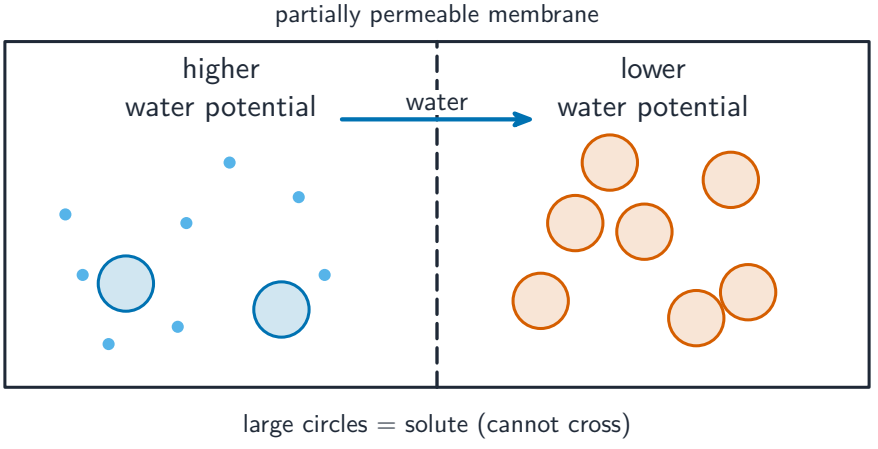
Charged **ions** and large polar molecules (such as **glucose**) cannot cross the oily bilayer by themselves. In **facilitated diffusion** they cross through channel proteins or carrier proteins, still moving down the concentration gradient. It is also passive.

Osmosis

Osmosis is the diffusion of water across a partially permeable membrane, from a higher **water potential** to a lower water potential. It is passive.



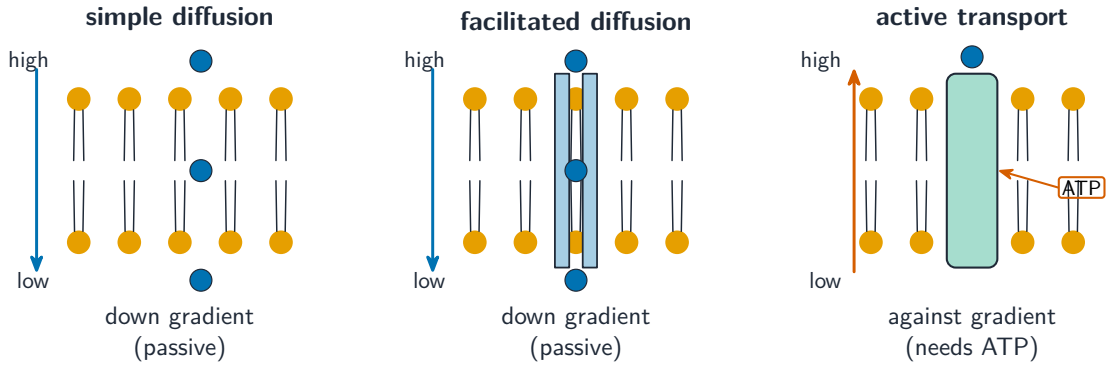
A potato osmometer: the sugar solution rises up the tube as water enters the potato by osmosis



Water crosses to the lower water potential; the solute is too big to cross the partially permeable membrane

Active transport

Active transport moves a substance **against** its concentration gradient – from low to high concentration. This needs carrier proteins and energy from ATP.

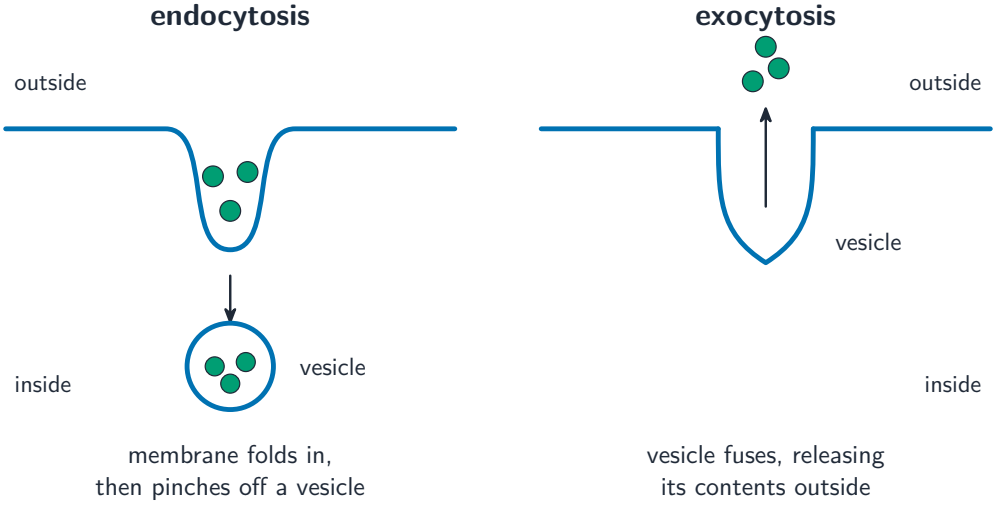


Diffusion and facilitated diffusion are passive (down the gradient); active transport goes against it and needs ATP

Endocytosis and exocytosis

These move large amounts of material in bulk, using ATP.

- in **endocytosis**, the membrane folds inwards around material and pinches off a **vesicle** to bring it into the cell.
- in **exocytosis**, a vesicle fuses with the membrane and releases its contents outside the cell.



Endocytosis brings material in by forming a vesicle; exocytosis fuses a vesicle to release its contents

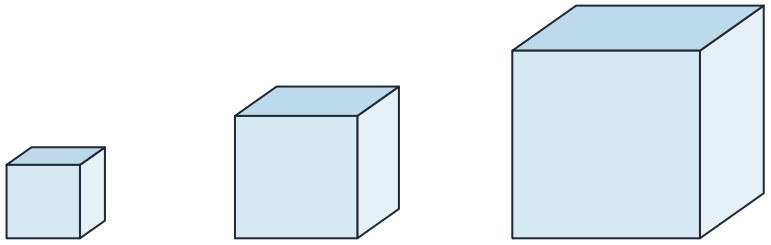
Surface area to volume ratio

A cell takes in and removes substances across its surface. As an object gets bigger, its **volume** grows faster than its **surface area**. So the surface area to volume ratio gets **smaller** as size increases.

For a cube of side L:

$$\text{surface area} = 6L^2, \qquad \text{volume} = L^3, \qquad \text{ratio} = \frac{6}{L}.$$

A large L gives a small ratio. This is why small cells (and thin, flat shapes) exchange materials quickly, while large cells cannot rely on diffusion alone.



side $L = 1$
SA : V $= 6 : 1$

side $L = 2$
SA : V $= 3 : 1$

side $L = 3$
SA : V $= 2 : 1$

a bigger cube (or cell) has a **smaller** surface area : volume ratio

As a cube (or cell) grows, its surface area : volume ratio gets smaller

> **Worked example.** Compare the surface area : volume ratio of a cube-shaped cell of side 4 with one of side 10.

For side 4: surface area $= 6 \times 4^2 = 96$ and volume $= 4^3 = 64$, so the ratio is $96 : 64 = 1.5 : 1$. For side 10: surface area $= 6 \times 10^2 = 600$ and volume $= 10^3 = 1000$, so the ratio is $600 : 1000 = 0.6 : 1$. The larger cell has the smaller ratio, so it exchanges materials across its surface more slowly for its size – which is why large organisms need specialised exchange surfaces such as lungs and gills.

You can show this with **agar** blocks of different sizes soaked in dye or acid: the smallest block, with the largest surface area to volume ratio, changes colour all the way through fastest. Diffusion across non-living materials can also be studied with **dialysis tubing** (Visking tubing).

Water potential and living cells

Water potential measures how likely water is to leave a solution. Pure water has the highest water potential. Adding a **solute** (a dissolved substance) lowers it. Water always moves by osmosis from a higher to a lower water potential.

To estimate the water potential of plant tissue, you place pieces in sucrose solutions of different water potentials. The solution that causes **no change** in mass or length has about the same water potential as the tissue.

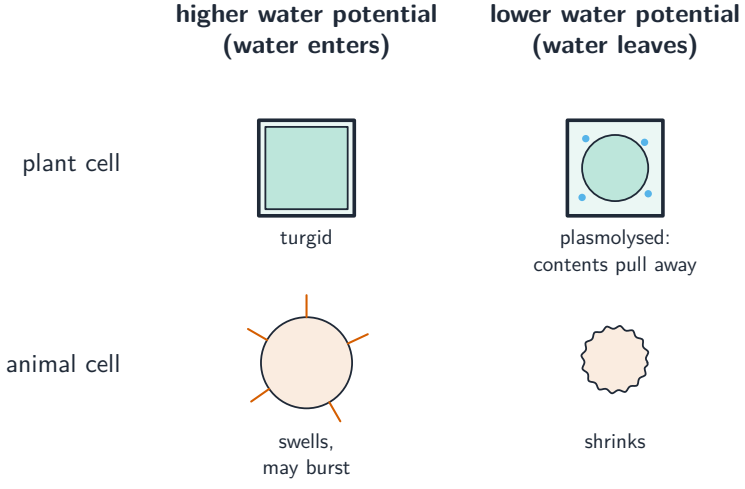
Effect on plant cells

- in a solution of **higher** water potential (for example **distilled water**), water enters the cell. The cell swells and becomes **turgid**, but the strong cell wall stops it bursting.
- in a solution of **lower** water potential, water leaves. The cell contents shrink and the membrane pulls away from the cell wall – this is **plasmolysis**.

Effect on animal cells

Animal cells have no cell wall to protect them.

- in a solution of higher water potential, water enters and the cell may burst. In a red blood cell this bursting is called **haemolysis**.
- in a solution of lower water potential, water leaves and the cell shrinks.



A plant cell becomes turgid or plasmolysed; an animal cell may burst (haemolysis) or shrink

Exam tips

- Describe the membrane as a **fluid mosaic**: a phospholipid bilayer with proteins, cholesterol and glycoproteins.
- Sort each process: diffusion, facilitated diffusion and osmosis are **passive** (down a gradient); active transport and endo/exocytosis need **ATP** and can go against it.
- For osmosis always use **water potential** (Ψ): water moves from **high (less negative) to low (more negative)** Ψ; pure water is 0, the highest.
- State the outcome by cell type: plant cell turgid/plasmolysed; animal cell lyses/crenates – link it to the water-potential gradient.

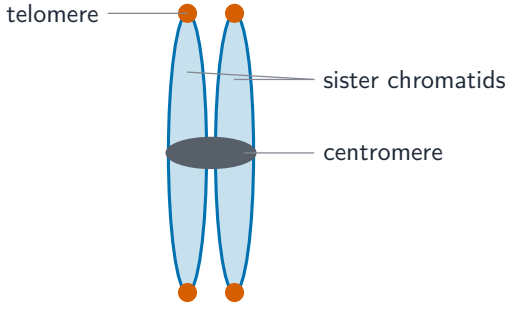
The mitotic cell cycle

A-Level Biology

The structure of a chromosome

A **chromosome** is one very long molecule of DNA wound tightly around special **proteins** called **histones**. Winding the DNA like this lets a huge length fit inside the nucleus and keeps it tidy.

Before a cell divides, its DNA is copied (this copying is called **replication**). After copying, each chromosome is made of two identical copies joined together. These two copies are the **sister chromatids**, and they are held together at a point called the **centromere**. The tips of each chromosome are capped by **telomeres**, which protect the ends.



one chromosome = two identical copies after DNA replication

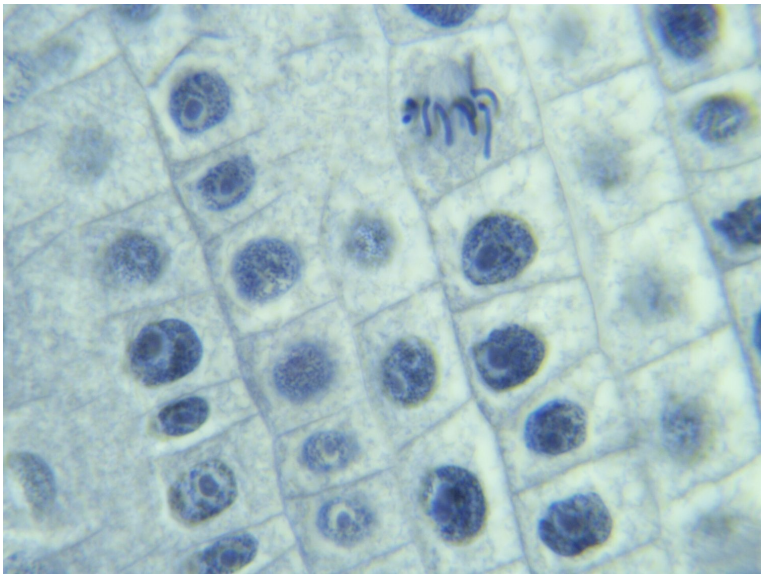
After replication a chromosome is two sister chromatids joined at the centromere, with telomeres at the tips

Why mitosis matters

Mitosis is a type of nuclear division that makes two **daughter cells** that are **genetically identical** – they carry exactly the same **genes** as the parent cell and as each other.

This matters for:

- **growth** of **multicellular** **organisms** (making more cells).
- **replacement** of damaged or dead cells.
- **repair** of **tissues** by making new cells.
- **asexual reproduction** (one parent makes identical offspring).

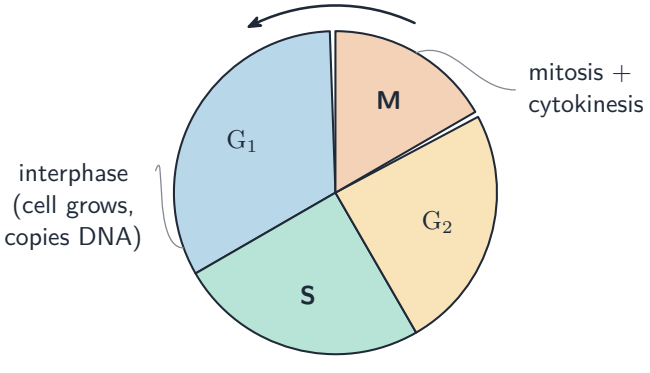


Onion root tip at metaphase: mitosis produces identical body cells for growth and repair

The mitotic cell cycle

The **cell cycle** is the full life of a cell from one division to the next. It has three parts:

1. **interphase** – the longest part. The cell grows in the $\mathbf{G_1}$ phase, copies its DNA in the **S** phase (replication), and grows again and prepares to divide in the $\mathbf{G_2}$ phase.
2. **mitosis** – the nucleus divides into two identical nuclei.
3. **cytokinesis** – the rest of the cell splits, giving two separate daughter cells.



Most of the cycle is interphase (G_1, S, G_2); mitosis (M) and cytokinesis are a short part

The mitotic index

The mitotic index is the fraction of cells in a sample that are in mitosis. You find it by counting cells under a microscope:

$$\text{mitotic index} = \frac{\text{number of cells in mitosis}}{\text{total number of cells}}.$$

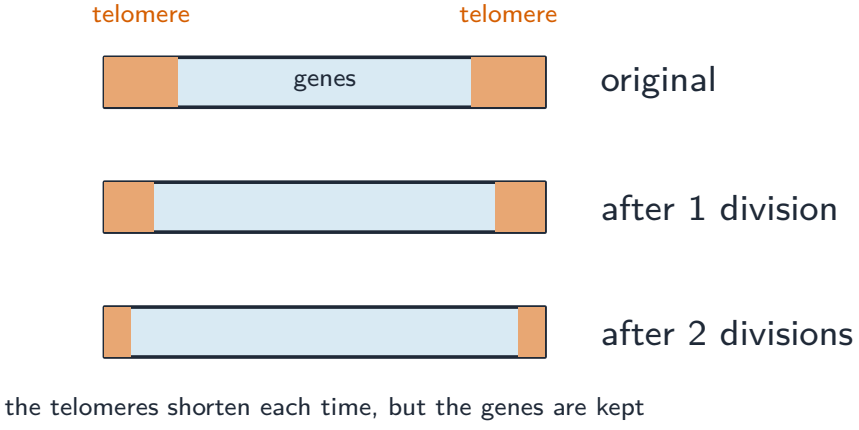
> **Worked example.** In a root-tip sample, 8 of the 50 cells seen are in mitosis. Find the mitotic index.

$$\text{mitotic index} = \frac{8}{50} = 0.16.$$

A high mitotic index means many cells are dividing quickly – normal in a growing root tip, but in adult animal tissue it can be a warning sign of uncontrolled growth.

Telomeres and the ends of chromosomes

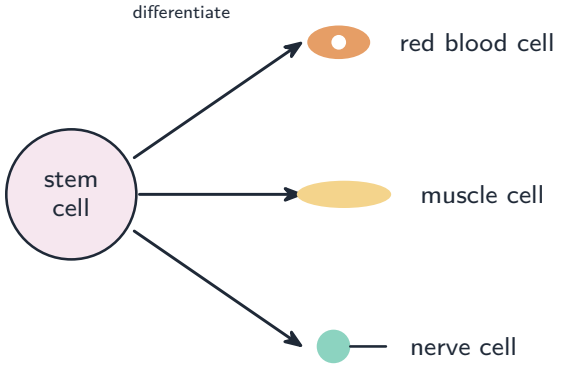
When DNA is replicated, the copying cannot reach the very end of the molecule, so a little is lost each time. Telomeres are short, repeated lengths of DNA at the ends that carry **no genes**. Because the telomeres are shortened instead, no important genes are lost during replication.



The telomeres (which carry no genes) shorten a little at each division, so the genes in the middle are never lost

Stem cells

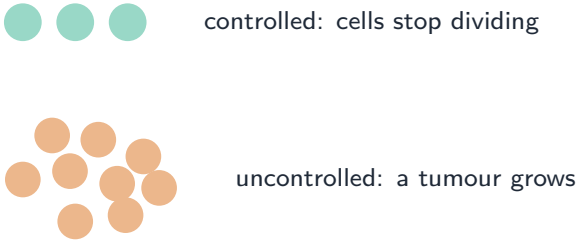
A **stem cell** is an unspecialised cell that can keep dividing by mitosis and can **differentiate** (change) into different specialised cell types. Stem cells are the source of new cells for replacing lost cells and repairing tissues.



One unspecialised stem cell can differentiate into many specialised cell types

Uncontrolled division and tumours

The cell cycle is normally tightly controlled, so cells divide only when needed. If this control is lost, a cell may divide again and again without stopping. This uncontrolled division produces a lump of cells called a **tumour** .

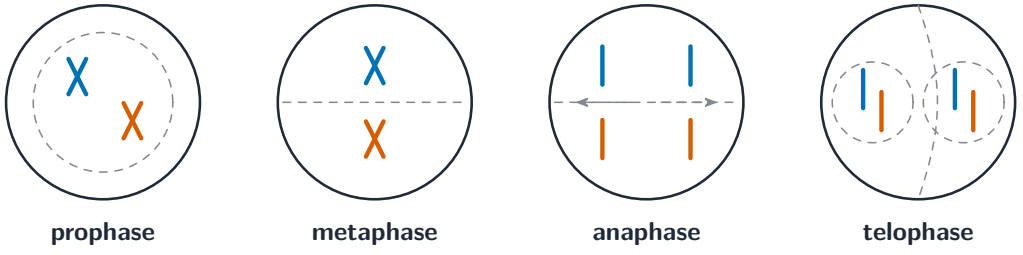


Normally cells stop dividing when they should; if that control is lost, they keep dividing into a tumour

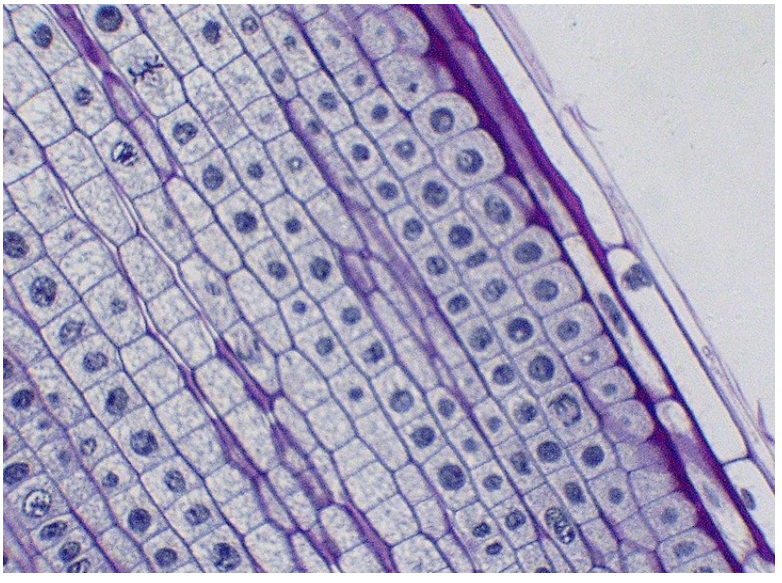
The stages of mitosis

Mitosis runs through four stages. You should be able to recognise them in photomicrographs and slides.

Stage	What happens
prophase	chromosomes coil up and become visible as two sister chromatids; the **nuclear envelope** breaks down; a **spindle** of fibres forms across the cell
metaphase	chromosomes line up along the middle (the **equator**); spindle fibres attach to each centromere
anaphase	the centromeres split; the sister chromatids are pulled to opposite ends (poles) of the cell
telophase	a set of chromosomes reaches each pole; a new nuclear envelope forms around each set, making two nuclei



The four stages: prophase, metaphase, anaphase, telophase



A real onion root tip: in a growing tip many cells are caught dividing

Cytokinesis then follows. In an animal cell the cell surface membrane pinches inwards to split the cell; in a plant cell a new wall forms across the middle. The result is two genetically identical daughter cells.

Exam tips

- Learn the stages **in order** (prophase, metaphase, anaphase, telophase) with one key event each; interphase (G1, S, G2) is **not** part of mitosis.
- Mitosis gives **two genetically identical diploid** cells – for growth, repair and asexual reproduction.
- **Mitotic index** = cells in mitosis ÷ total cells; a high value means rapid division.
- Link **uncontrolled mitosis** to tumours and telomere shortening to the limit on the number of divisions.

Nucleic acids and protein synthesis

A-Level Biology

Nucleotides – the building blocks

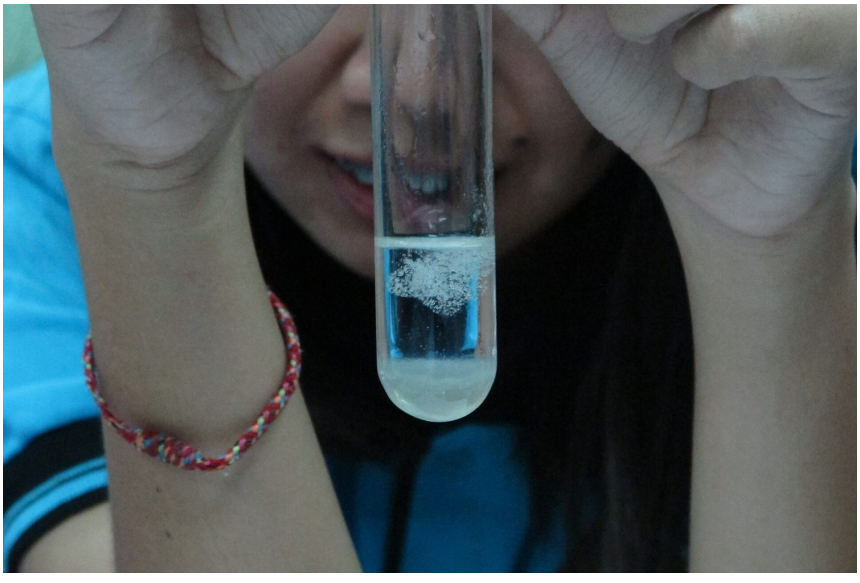
Nucleic acids (DNA and RNA) are polymers of small units called **nucleotides** . Each nucleotide is made of three parts joined together:

- a **phosphate** group,
- a sugar (a 5-carbon sugar),
- a nitrogen-containing **base** .

ATP is a special nucleotide. It has the base **adenine** , the sugar **ribose** , and **three** phosphate groups. Breaking off the last phosphate releases **energy** for the cell.



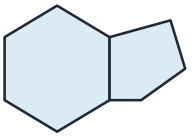
A nucleotide is a phosphate, a sugar and a base; ATP is a nucleotide with three phosphates



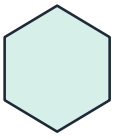
DNA extracted from cells appears as pale, stringy strands of these nucleotide polymers

There are five bases, in two groups:

- **purines** have a **double ring** (two rings): adenine and **guanine** .
- **pyrimidines** have a **single ring** (one ring): **cytosine** , **thymine** and **uracil** .



purine
double ring (A, G)



pyrimidine
single ring (C, T, U)

The two groups differ in shape: a purine is a double ring, a pyrimidine a single ring – which is why a purine always pairs with a pyrimidine

The structure of DNA

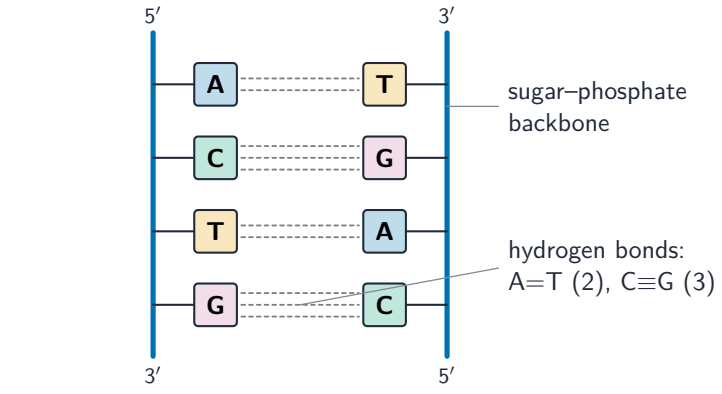
A DNA molecule is two strands twisted together into a **double helix** .

Each **strand** has a backbone of alternating sugar (here the sugar is **deoxyribose**) and phosphate. The sugar of one nucleotide is joined to the phosphate of the next by a **phosphodiester bond** .

The two strands are held together by their bases, which meet in the middle. The pairing is exact – this is **complementary base pairing** :

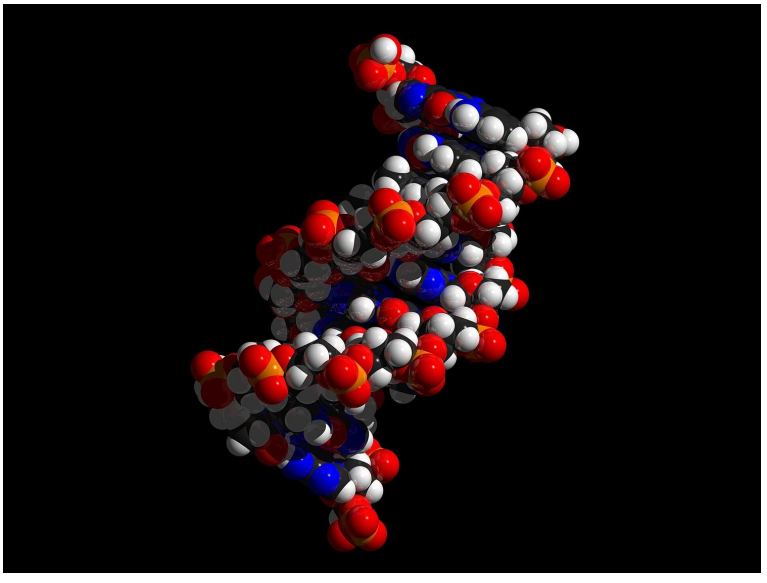
- A always pairs with T, held by **two hydrogen bonds** .
- C always pairs with G, held by **three** hydrogen bonds (so a C–G **base pair** is harder to separate).

The two strands run in opposite directions: one goes 5′ to 3′ while the other goes 3′ to 5′. We say they are **antiparallel** .



Complementary base pairing: A pairs with T (2 hydrogen bonds), C with G (3); the strands are antiparallel

The flat ladder above is twisted into a spiral. This space-filling model, where every atom is a ball, shows the real shape of the **double helix**:



*A space-filling model of DNA: the two strands twist around each other into the **double helix** – the ladder of the diagram, coiled up*

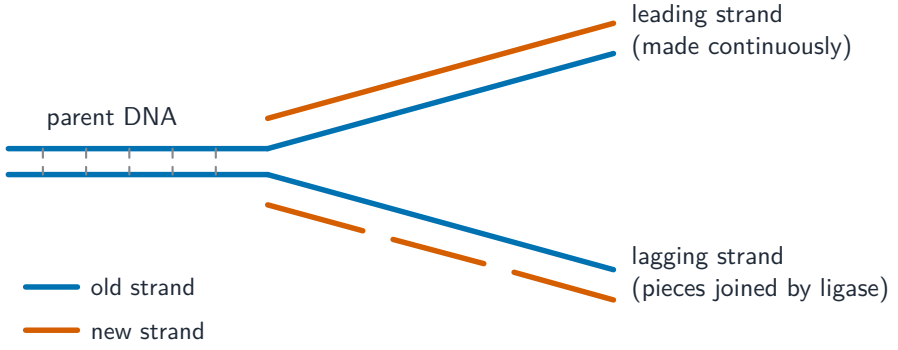
DNA replication

DNA **replication** (copying) happens during the S phase of the cell cycle. It is **semi-conservative**: each new molecule keeps one old strand and one new strand. The steps are:

1. the double helix unwinds and the hydrogen bonds break, so the two strands separate.
2. each old strand acts as a template. Free nucleotides pair with the exposed bases by complementary base pairing.
3. the **enzyme DNA polymerase** joins the new nucleotides into a strand. It can only add nucleotides in the 5′ to 3′ direction.

Because of that 5′ to 3′ rule, the two new strands are made differently:

- the **leading strand** is built continuously, following the unwinding.
- the **lagging strand** is built in short pieces, working away from the unwinding point. The enzyme **DNA ligase** then joins these pieces together.

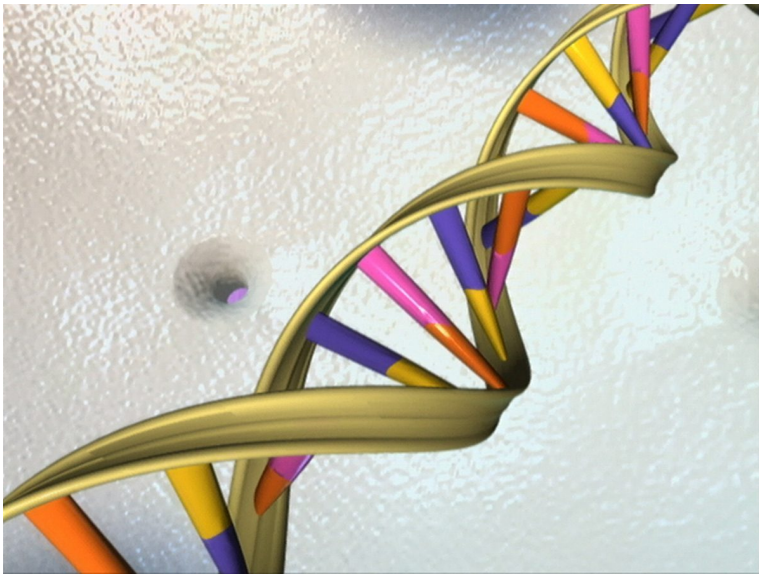


each new molecule = one old strand + one new strand (semi-conservative)

Replication is semi-conservative: each new molecule keeps one old strand (blue) and one new strand (orange)

RNA

RNA is also made of nucleotides, but it is a single strand, its sugar is ribose, and it uses uracil in place of thymine. The most important type here is **messenger RNA (mRNA)**, which carries a copy of a gene's instructions out of the nucleus to be used.



DNA double helix: RNA is a related nucleic acid that carries genetic information into protein synthesis

The genetic code

A **gene** is a sequence of DNA nucleotides that codes for one **polypeptide**.

The code is read in **triplets** – groups of three bases. Each triplet either codes for one specific **amino acid**, or acts as a start or stop signal. The code is **universal**: nearly all living things use the same triplets for the same amino acids.

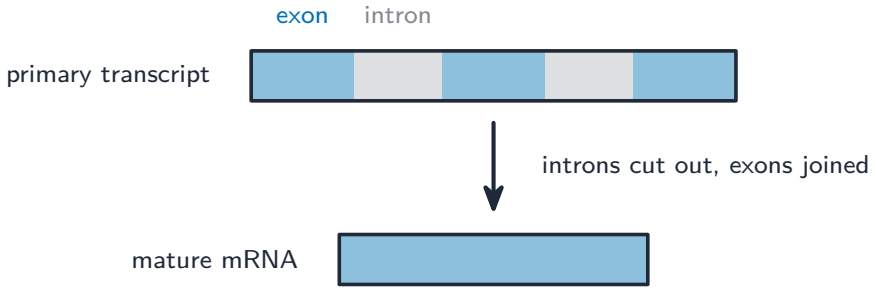
There are 64 possible triplets but only 20 common amino acids, so most amino acids are coded by more than one triplet.

Protein synthesis: transcription and translation

Transcription (in the nucleus)

The DNA gene is copied into mRNA. This is **transcription**.

- the strand of DNA that is copied is the **template strand**; the partner strand is the **non-transcribed strand**.
- the enzyme **RNA polymerase** joins RNA nucleotides that pair with the template bases (with uracil pairing to adenine).
- in eukaryotes the first RNA made (the **primary transcript**) contains coding parts called **exons** and non-coding parts called **introns**. The introns are cut out and the exons joined together to form the finished mRNA.

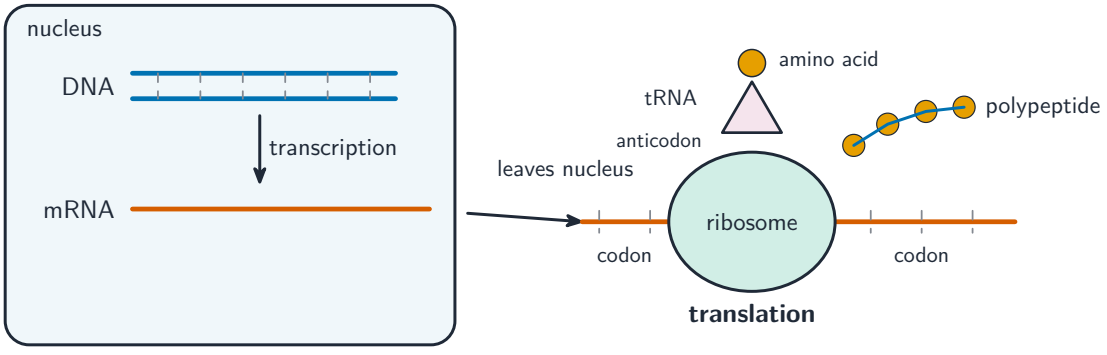


Splicing: the non-coding introns are cut out and the coding exons joined to make the finished mRNA

Translation (at the ribosome)

The mRNA leaves the nucleus and attaches to a **ribosome** . Building the polypeptide from the mRNA code is **translation** .

- the mRNA is read in **codons** (each codon is one triplet of mRNA bases).
- molecules of **transfer RNA (tRNA)** bring amino acids to the ribosome. Each tRNA has an **anticodon** that pairs with a matching codon.
- as the codons are read in order, the ribosome joins the amino acids with **peptide bonds** , building the polypeptide.



Transcription copies DNA into mRNA in the nucleus; translation at the ribosome builds the polypeptide

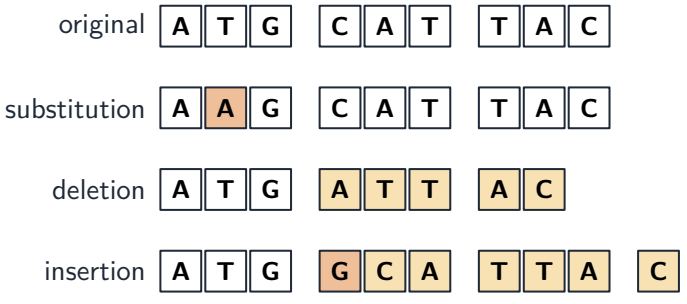
Worked example. A DNA template strand reads `TAC GGA CTT`. Give the mRNA codons, and say how long the polypeptide is. Transcribe by **complementary base pairing**, remembering that RNA uses **uracil** in place of thymine: `TAC` gives `AUG`, `GGA` gives `CCU`, `CTT` gives `GAA`. So the mRNA reads `AUG CCU GAA`. `AUG` is the **start** codon (methionine), so the polypeptide is **three** amino acids long - two once the start methionine is removed. Two errors cost most of the marks here: pairing A with **T** instead of **U** when writing mRNA, and transcribing the **coding** strand instead of the **template** strand. The mRNA is complementary to the template, and identical to the coding strand apart from U replacing T.

Gene mutations

A gene **mutation** is a change in the base sequence of a DNA molecule. It may change the polypeptide made. There are three types:

- **substitution** – one base is swapped for a different base. This changes at most one amino acid, and sometimes none (because most amino acids have more than one triplet).
- **deletion** – a base is removed.
- **insertion** – an extra base is added.

A deletion or insertion shifts how every later triplet is read, so it usually changes many amino acids after that point and has a large effect on the polypeptide.



A substitution changes one triplet; a deletion or insertion shifts every later triplet (a frameshift)

Exam tips

- DNA is **antiparallel**, held by hydrogen bonds (A=T two, C≡G three); replication is **semi-conservative** (each product keeps one old strand).
- State **where** and **what**: transcription (nucleus → mRNA), translation (ribosome, mRNA + tRNA → polypeptide).

- Describe the code with all four terms: **triplet, non-overlapping, degenerate, universal**.
- For a **gene mutation** (substitution, insertion, deletion) explain the effect on the protein – and why a substitution can be silent (degeneracy).

Transport in plants

A-Level Biology

The two transport tissues

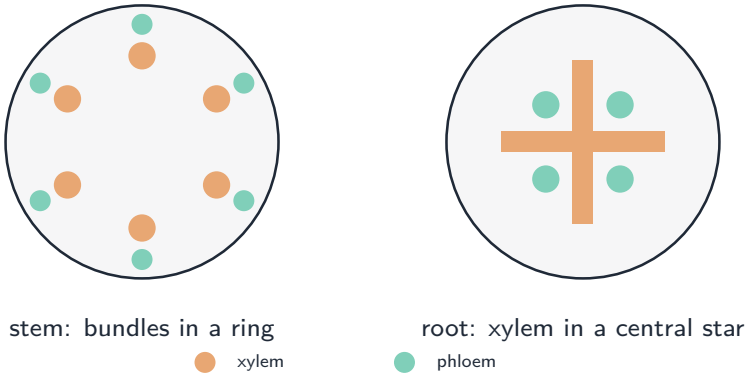
Plants move substances through two transport **tissues** :

- **xylem** carries water and dissolved **mineral ions** up from the roots.
- **phloem** carries dissolved foods (called **assimilates** , mainly sugars) to wherever they are needed.

In a **transverse section** (a cut straight across) of a **dicotyledonous plant** :

- in the **stem**, xylem and phloem sit together in bundles near the outside, with xylem on the inside of each bundle.
- in the **root**, the xylem is in the centre, often in a star shape, with phloem between the arms.
- in the **leaf**, both are found in the veins.

When you draw a plan diagram, you draw only the outlines of the tissues, not the single cells.



Where the tissues sit: stem bundles form a ring near the edge (xylem inside); in the root the xylem makes a central star



The wood of a tree trunk is xylem; each ring is one year of growth

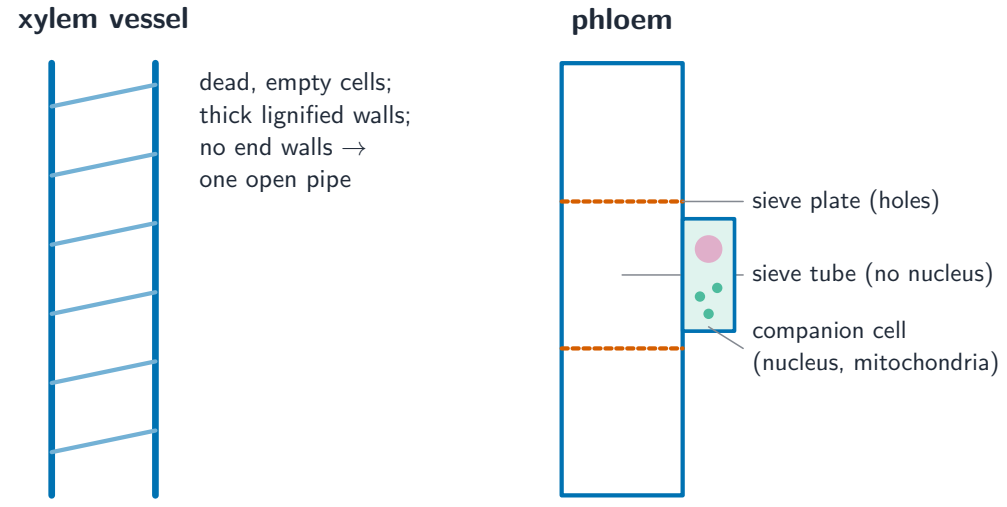
Xylem vessels

Xylem water-carrying tubes are called **vessels** . They are made of dead, empty cells joined end to end, with the end walls gone, so they form one long open pipe. Their walls are thickened and waterproofed with **lignin** . So the structure suits the job: the hollow, open tube with no contents lets water flow fast, and the lignin gives strength and support.

Phloem sieve tubes and companion cells

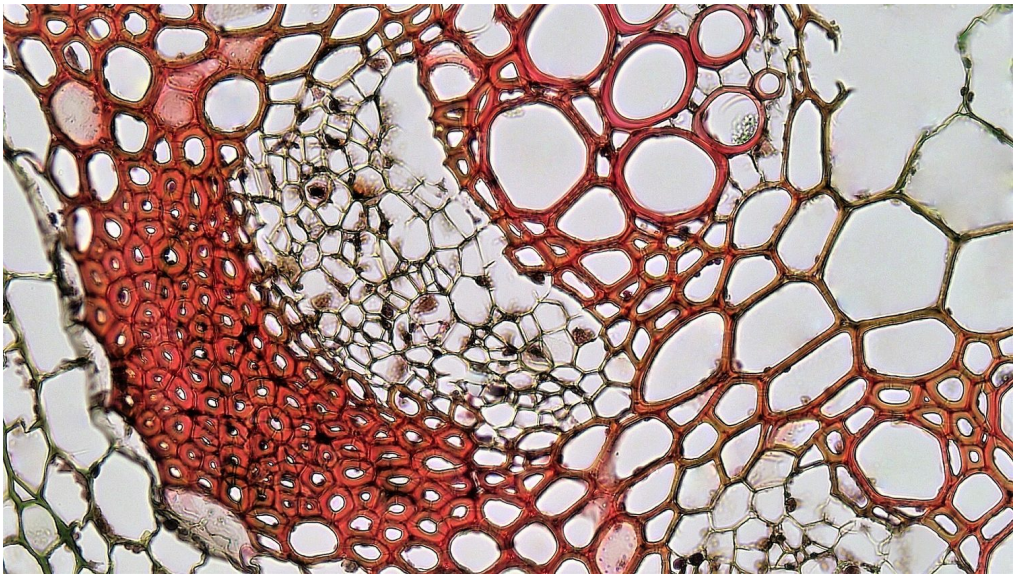
Phloem food-carrying tubes are called **sieve tubes** . They are living cells joined end to end, but their end walls are not gone – they become **sieve plates** with many holes that sap flows through. To leave room for flow, a sieve tube cell loses most of its contents and has no nucleus.

Beside each sieve tube is a **companion cell** . It keeps its **nucleus** and has many **mitochondria** . It does the living work for the sieve tube and loads sugars into it.



Xylem is a dead, open pipe; phloem is living sieve tubes with sieve plates and companion cells

This is what a real **vascular bundle** looks like under the microscope, in a stained section of a young sunflower stem:



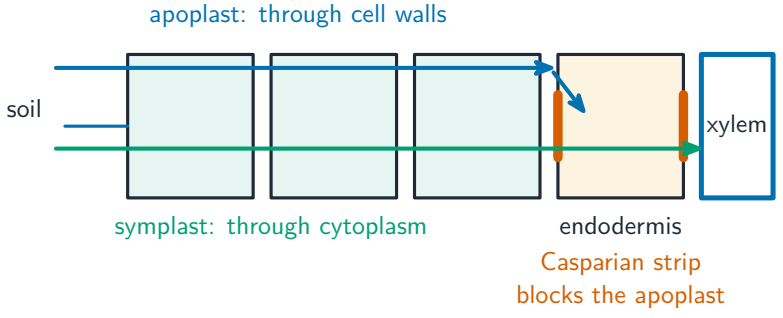
*A real **vascular bundle** in section: the big red cells are xylem vessels (thick lignified walls); the smaller cells above are phloem*

Water from the soil to the xylem

Water enters a **root hair cell** by **osmosis**, because the root hair has a lower water potential than the soil water. Water then crosses the root to the xylem by two pathways:

- the **apoplast pathway** – water moves through the **cell walls** (made of **cellulose**) and the spaces between cells, without entering the cytoplasm. This is fast.
- the **symplast pathway** – water moves through the cytoplasm of cells, passing from cell to cell through the **plasmodesmata**.

At a ring of cells called the **endodermis**, the apoplast pathway is blocked by the **Casparian strip**, a waterproof band of **suberin**. This forces all the water through the cell membranes, which lets the plant control what enters the xylem.



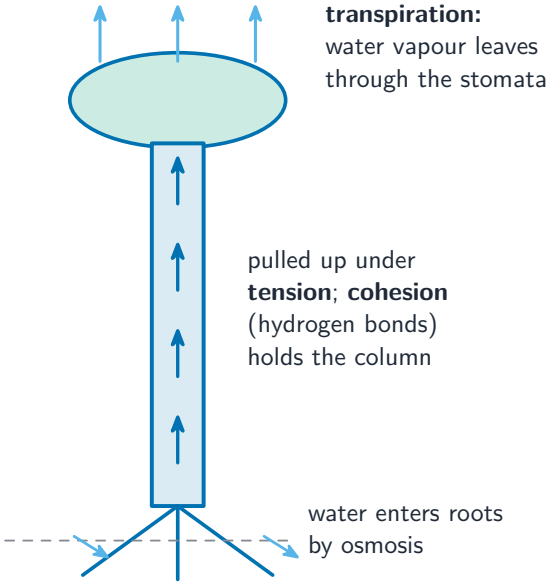
The apoplast goes through the walls, the symplast through the cytoplasm; the Casparian strip blocks the apoplast at the endodermis

Transpiration and the movement of water up the xylem

Transpiration is the loss of water vapour from a plant. Water **evaporates** (turns to vapour) from the wet cell surfaces inside the leaf – this is **evaporation**. The **water vapour** then **diffuses** out through the **stomata** into the **atmosphere**.

This loss at the top pulls water up the xylem in a continuous column. It works because of hydrogen bonding between water molecules:

- water molecules attract each other through **hydrogen bonds**, so they stick together. This sticking is **cohesion**, and it lets the whole column be pulled up under **tension** (the cohesion–tension idea).
- water molecules also stick to the cellulose of the cell walls. This is **adhesion**, which helps hold the column in place.

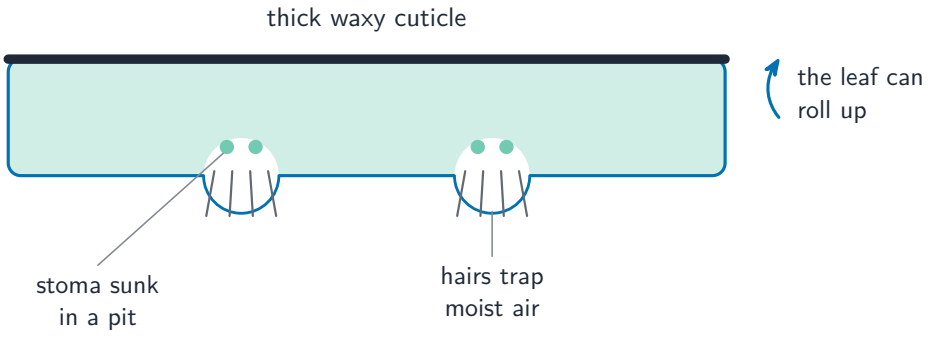


Transpiration at the leaf pulls the whole water column up the xylem; cohesion (hydrogen bonds) keeps it together

Worked example. A plant cell with a water potential of −800 kPa is placed in a solution of water potential −400 kPa. Which way does water move, and what happens to the cell? Water always moves **down** a water potential gradient - from the **less negative** (higher) value to the **more negative** (lower) one. The solution at −400 is **higher** than the cell at −800, so water moves **into** the cell. The cell swells, the protoplast presses on the cell wall, the **pressure potential** rises, and the cell becomes **turgid**. Two traps: −400 is **greater** than −800, which is the sign error that reverses half of all answers; and pure water is the maximum at **0**, so every solution is negative and a water potential can never rise above zero.

Xerophytes

A **xerophyte** is a plant **adapted** to live where water is scarce. Its leaves reduce water loss by transpiration in several ways: a thick waxy **cuticle**, **stomata** sunk in pits, hairs that trap moist air, and leaves that can roll up. You should be able to draw a labelled leaf section showing these features.



Xerophyte leaves cut water loss: a thick cuticle, sunken stomata, trapped moist air and rolling up

Translocation: moving assimilates in the phloem

Assimilates such as **sucrose** and **amino acids** are carried in the phloem from a **source** to a **sink** .

- a **source** is where the assimilate is made or released (for example a photosynthesising leaf).
- a **sink** is where it is used or stored (for example a growing root).

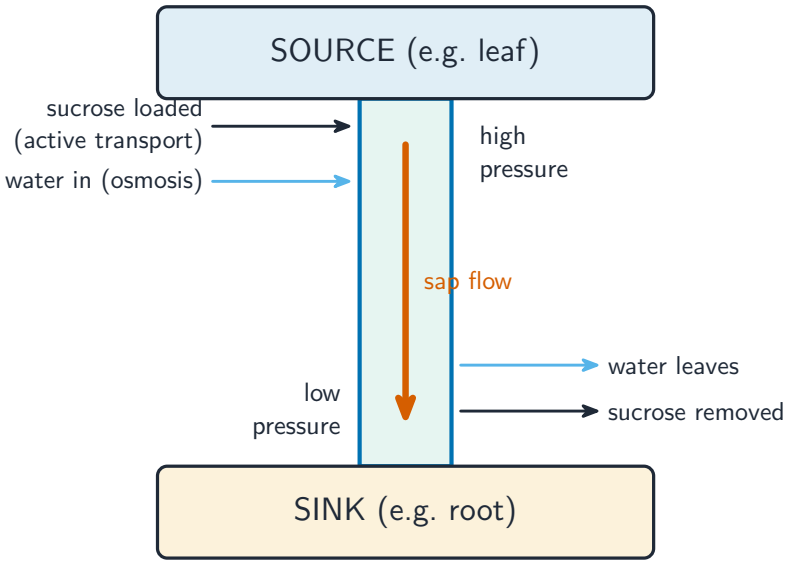
■ Loading at the source

Companion cells load sucrose into the sieve tubes against its concentration gradient. They use **proton pumps** to pump hydrogen ions out, then **cotransporter proteins** bring sucrose back in together with those ions. This is a form of **active transport** .

Loading sucrose lowers the **water potential** inside the sieve tube, so water follows by osmosis. This raises the **hydrostatic pressure** there.

■ Mass flow

At the sink, sucrose is removed, so the water potential rises, water leaves, and the pressure falls. The result is a pressure difference between source (high) and sink (low). Sap flows from high to low pressure down this gradient. This pressure-driven flow is called **mass flow** .



Loading sucrose at the source raises the pressure; sap then flows by mass flow to the sink

Exam tips

- State the tissue for each direction: **xylem** carries water up (dead, lignified), **phloem** carries assimilates both ways (living sieve tubes + companion cells).
- Explain water movement by **cohesion-tension**: transpiration pulls a continuous water column held by cohesion (H-bonds) and adhesion.
- List the factors affecting **transpiration rate** (light, temperature, humidity, air movement) and how a potometer measures uptake.
- For **xerophytes**, link each adaptation (thick cuticle, sunken stomata, rolled leaves, hairs) to reduced water loss.

Transport in mammals

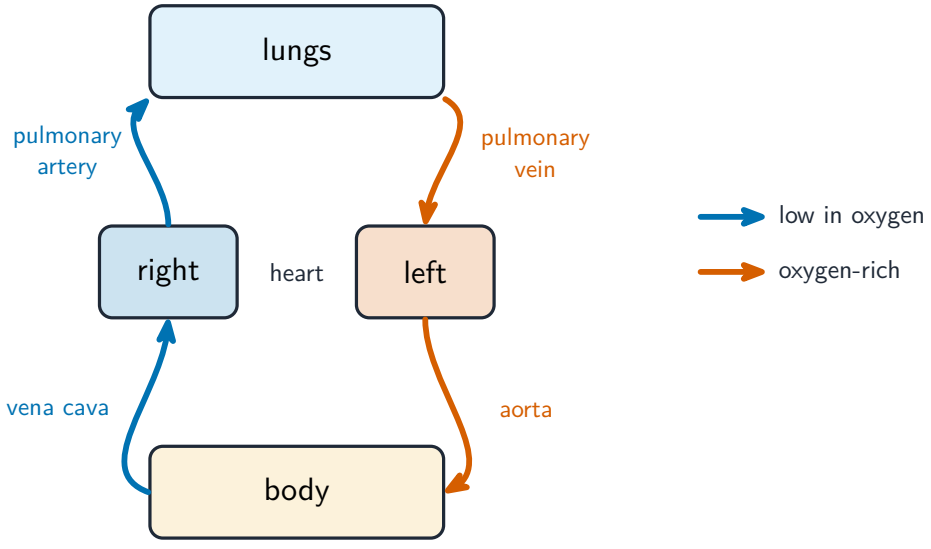
A-Level Biology

The circulatory system

Mammals have a **closed double circulation**. "Closed" means the blood stays inside **blood vessels** the whole time. "Double" means the blood passes through the **heart** twice for each full trip around the body. This gives two linked loops, so we call it a **double circulation** and the whole thing a **circulatory system** :

- the **pulmonary circulation** carries blood from the heart to the lungs and back.
- the **systemic circulation** carries blood from the heart to the rest of the body and back.

Blood travels in this order: **arteries** → **arterioles** → **capillaries** → **venules** → **veins** .



Double circulation: the pulmonary loop goes to the lungs, the systemic loop to the body; blood passes through the heart twice

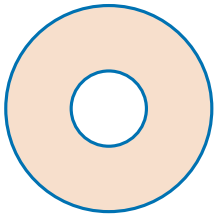
The main vessels are:

- the **pulmonary artery** – carries blood low in **oxygen** from the heart to the lungs.
- the **pulmonary vein** – carries oxygen-rich blood from the lungs back to the heart.

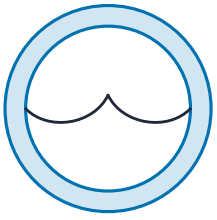
- the **aorta** – the big artery that carries oxygen-rich blood from the heart to the body.
- the **vena cava** – the big vein that returns oxygen-poor blood from the body to the heart.

How the vessels suit their jobs

Vessel	Structure	Function
artery	thick wall of **muscle** and **elastic** fibres; narrow **lumen** (the space inside)	carries blood at high pressure away from the heart; elastic walls stretch and recoil to smooth the flow; muscle controls the flow
capillary	wall just one cell thick; very narrow	short distance for exchange of substances between blood and cells
vein	thin wall; wide lumen; has **valves**	returns blood at low pressure to the heart; valves stop blood flowing backwards



artery
thick muscular, elastic wall; narrow lumen (high pressure)



vein
thin wall; wide lumen; valves (low pressure)

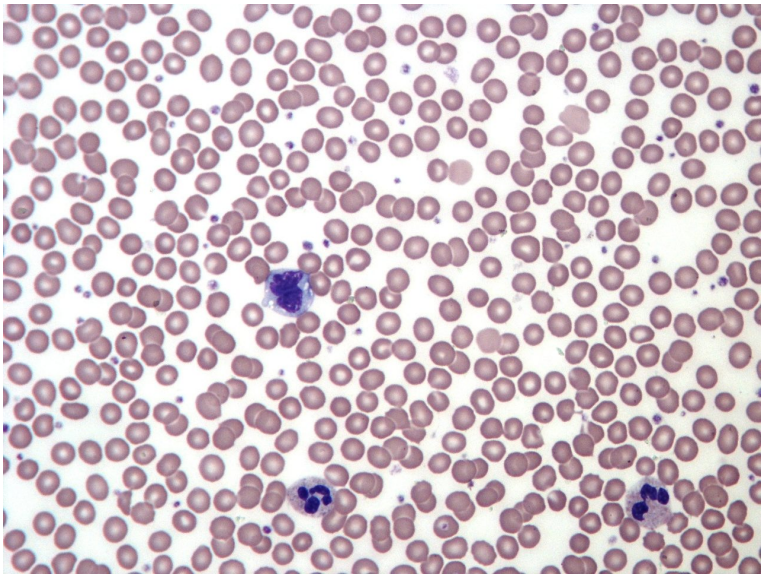


capillary
wall one cell thick; very narrow (exchange)

An artery has a thick wall and narrow lumen; a vein a thin wall, wide lumen and valves; a capillary is one cell thick

Blood cells

You should recognise: **red blood cells** (which carry oxygen), and three white blood cells – **monocytes** , **neutrophils** and **lymphocytes** .

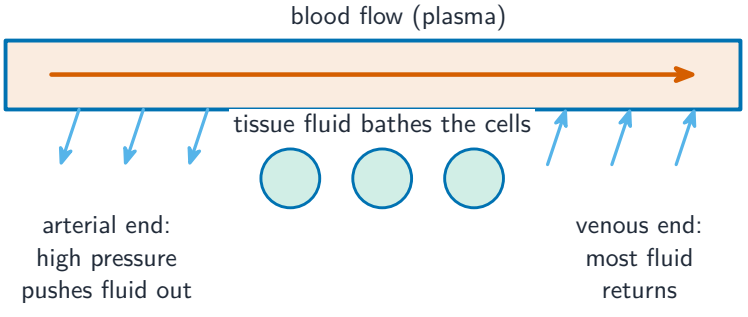


*A stained **blood smear**: many small red cells, plus a lymphocyte (centre) and neutrophils (lobed nucleus) – the white cells are larger and have a nucleus*

Water, plasma and tissue fluid

Water is the main part of blood. It is a good **solvent** , so it carries dissolved substances, and it has a high **specific heat capacity** , so the blood's temperature stays steady.

At the start of a capillary, the high blood pressure pushes liquid (but not the cells or large proteins) out of the **plasma** and through the capillary wall. This liquid around the cells is **tissue fluid** . It supplies the cells with oxygen and glucose and carries waste away. Most of it returns to the capillary at the far end, where the pressure is lower.



High pressure at the arterial end pushes fluid out to form tissue fluid; most returns at the venous end

Transport of oxygen and carbon dioxide

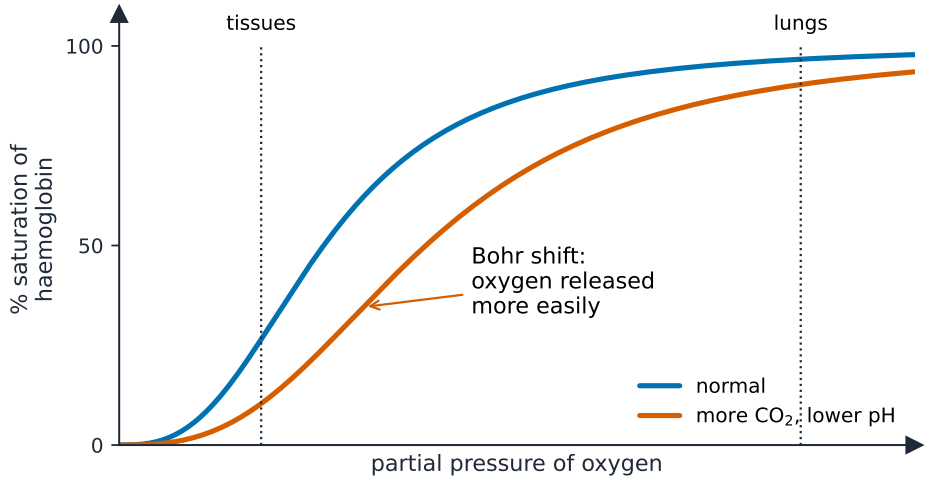
Carrying oxygen

Oxygen is carried by **haemoglobin** in the red blood cells. We show how well haemoglobin holds oxygen with the **oxygen dissociation curve**. This S-shaped graph plots the **saturation** (how full of oxygen the haemoglobin is) against the **partial pressure** of oxygen:

- where the partial pressure of oxygen is **high** (in the lungs), haemoglobin loads up and becomes almost fully saturated.
- where it is **low** (in respiring tissues), haemoglobin unloads its oxygen for the cells to use.

The Bohr shift

When tissues are very active, they release more **carbon dioxide**, which lowers the pH. This makes haemoglobin release oxygen more easily, so the curve moves to the right. This helpful change is the **Bohr shift**: oxygen is given up exactly where it is most needed.



Bohr shift (right): ↑ CO_2/ ↓ pH → lower O_2 affinity → unload at tissues

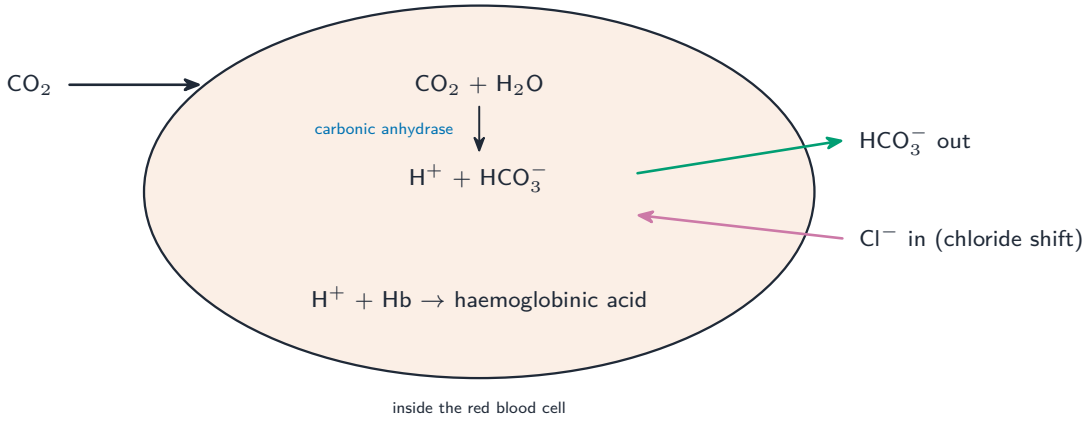
Haemoglobin loads oxygen in the lungs and unloads it in the tissues; the Bohr shift moves the curve right so more is released

Carrying carbon dioxide

A little carbon dioxide dissolves straight into the plasma, but most is carried after a reaction inside the red blood cells:

1. the **enzyme carbonic anhydrase** speeds up the reaction of carbon dioxide with water to make carbonic acid.
2. the carbonic acid splits into hydrogen ions and **hydrogencarbonate ions** .
3. the hydrogencarbonate ions move out into the plasma. This is the main way carbon dioxide is carried.
4. to keep the charge balanced, **chloride ions** move into the red blood cells. This movement is the **chloride shift** .
5. the hydrogen ions join haemoglobin to form **haemoglobinic acid** ; this mops up the hydrogen ions and keeps the pH steady.

Some carbon dioxide also joins haemoglobin directly to form **carbamino-haemoglobin** .



Most CO_2 travels as hydrogencarbonate ions; the chloride shift keeps the charge balanced and haemoglobin mops up the H^+

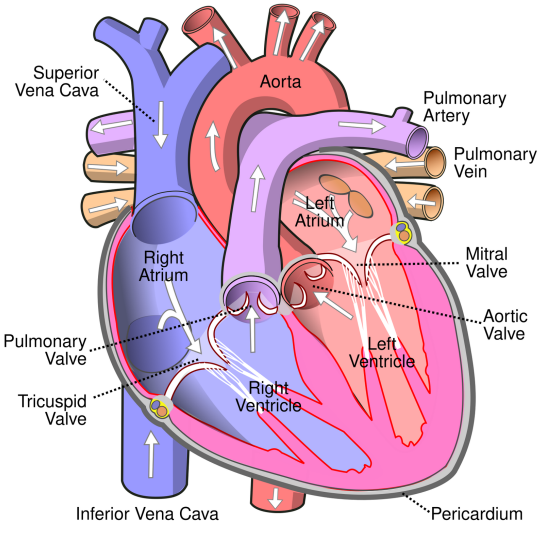
Worked example. At the lungs the partial pressure of oxygen is about 12 kPa and haemoglobin is about 98% saturated; in an exercising muscle it is about 3 kPa and saturation falls to about 40%. How much oxygen is unloaded, and why is the curve S-shaped? Subtract the saturations: $98 - 40 = \mathbf{58\%}$ of the haemoglobin's oxygen is released in the muscle. The **S shape** comes from **cooperative binding** - the first oxygen to bind changes haemoglobin's shape so the next ones bind more easily, which is why the middle of the curve is so steep. That steepness is the point: a **small** fall in partial pressure in a respiring tissue causes a **large** release of oxygen. The extra carbon dioxide in an exercising muscle lowers the pH and shifts the curve **right**, so even more is unloaded at the same partial pressure - that is the **Bohr shift** doing its job.

The heart

Structure

The heart has four chambers. The two upper chambers are the **atria** (singular: atrium); they have thin walls because they only push blood down into the chambers below. The two lower chambers are the **ventricles**; they have thick muscular walls because they pump blood out of the heart.

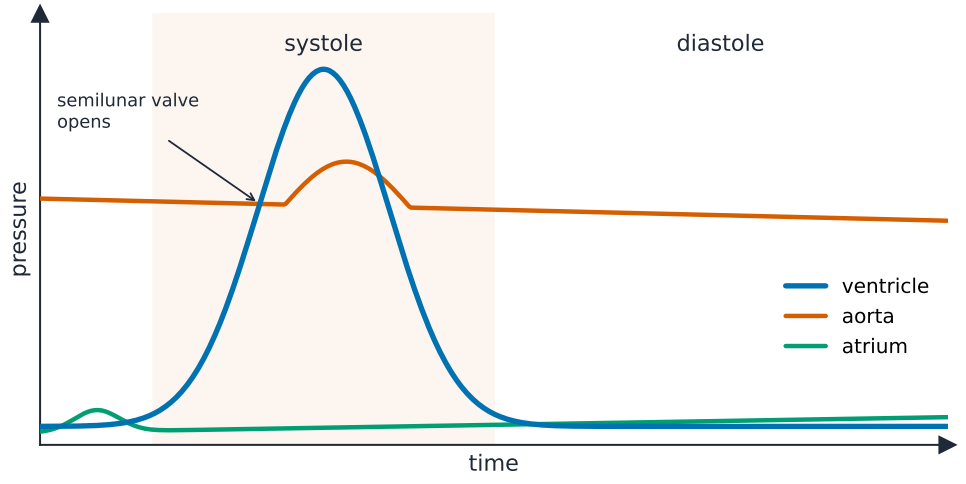
The left ventricle wall is thicker than the right ventricle wall, because the left side must pump blood all the way round the body, while the right side only pumps to the nearby lungs.



The four chambers, the valves and the main vessels; the left ventricle wall is the thickest

The cardiac cycle

One heartbeat is the **cardiac cycle** . It has two parts: **systole** (when heart muscle contracts) and **diastole** (when it relaxes and fills). When a chamber contracts, the pressure inside rises; this pressure change opens and closes the valves so that blood flows one way only.



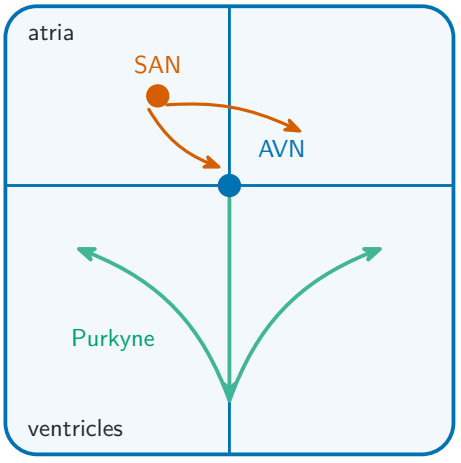
systole = contraction (high *P*) · diastole = relaxation/fill · valves open on *P* gradients

The ventricle pressure spikes in systole and pushes blood into the aorta; a valve opens or closes whenever two pressure curves cross

Controlling the heartbeat

The heart sets its own rhythm:

- the **sinoatrial node** in the right atrium is the pacemaker. It sends out a wave of electrical excitation that spreads across the atria and makes them contract.
- the **atrioventricular node** picks up the wave, holds it back for a moment (so the atria empty first), then passes it on.
- the **Purkyne tissue** carries the wave down and through the ventricle walls, so the ventricles contract from the bottom upwards and push blood out.



SAN fires → atria contract → AVN delays → Purkyne → ventricles contract from the bottom up

The SAN sets the rhythm; the wave passes to the AVN, then the Purkyne tissue makes the ventricles contract bottom-up

Exam tips

- Trace the **double circulation** and name the chambers, valves and vessels; the left ventricle wall is thicker (pumps further at higher pressure).
- Read the **cardiac-cycle** pressure graph: a valve opens/closes when the pressures across it cross over.
- Explain the **oxygen dissociation curve** (S-shaped) and the **Bohr shift** (more CO2 shifts it right, unloading more O2 to active tissue).
- Most CO2 is carried as **hydrogencarbonate** (via the chloride shift).

Gas exchange

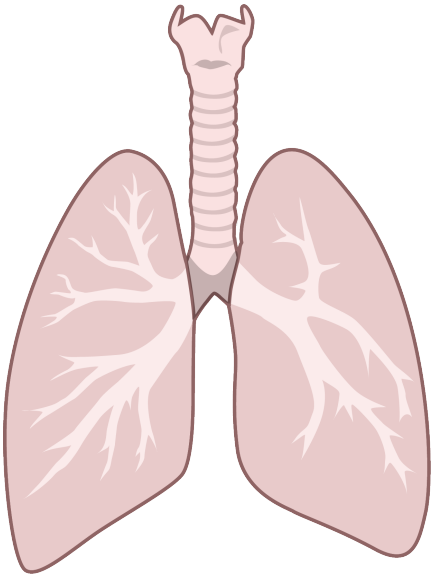
A-Level Biology

The gas exchange system

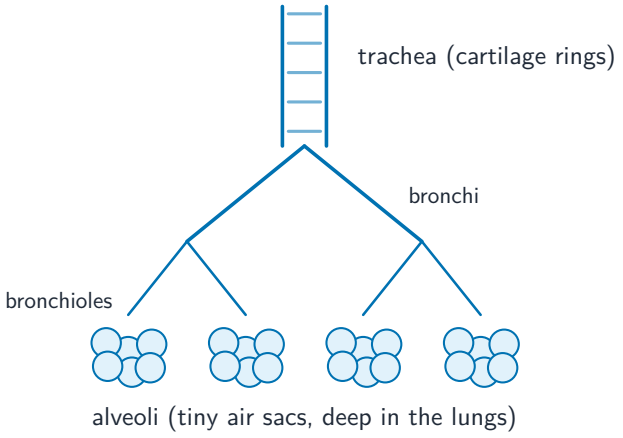
Your body needs to take in oxygen and get rid of carbon dioxide. This swap happens in the **gas exchange** system. Air follows this path into the body:

- down the **trachea** (the windpipe),
- into two **bronchi** (one to each lung),
- into many smaller **bronchioles** ,
- and finally into tiny air sacs called **alveoli** , deep in the **lungs** .

Each alveolus is wrapped in a network of **capillaries** , so air and blood are brought very close together.



The human lungs, where gas exchange takes place across millions of alveoli

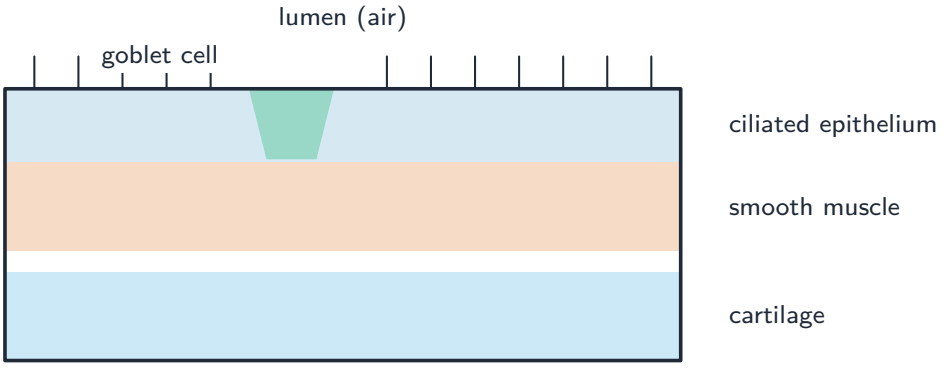


Air passes down the trachea, into the bronchi and bronchioles, to the alveoli

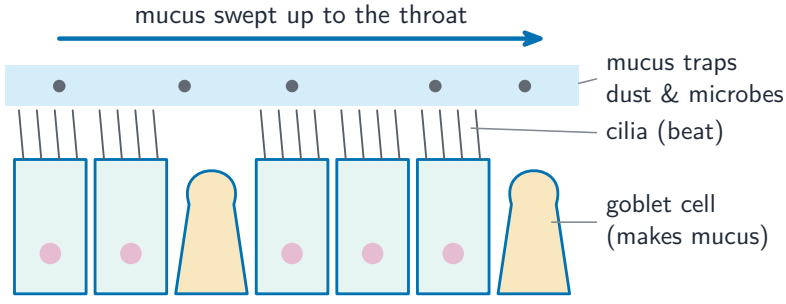
The tissues of the airways and what they do

Tissue	Where it is	Function
cartilage	C-shaped rings in the trachea and bronchi	holds the airway open so it cannot collapse when you breathe in
ciliated epithelium	lining the trachea and bronchi	tiny hairs called **cilia** beat to sweep **mucus** up towards the throat, away from the lungs
goblet cells and **mucous glands**	in the lining of the airways	make the mucus, which traps dust and **microbes** that you breathe in
smooth muscle	in the walls of bronchi and bronchioles	contracts to make the airway narrower
elastic fibres	in the airway and alveolus walls	stretch when you breathe in, then spring back to help push air out
squamous epithelium	the very thin, flat lining of the alveoli	gives a very short distance for gases to cross

The cilia, goblet cells and mucous glands work together to keep the lungs clean and healthy: the mucus traps dirt and microbes, and the cilia carry it away to be swallowed.



A section through the airway wall: cilia and goblet cells line the lumen, with smooth muscle and supporting cartilage behind



Goblet cells make mucus that traps dust and microbes; the cilia sweep it up to the throat

Gas exchange in the alveoli

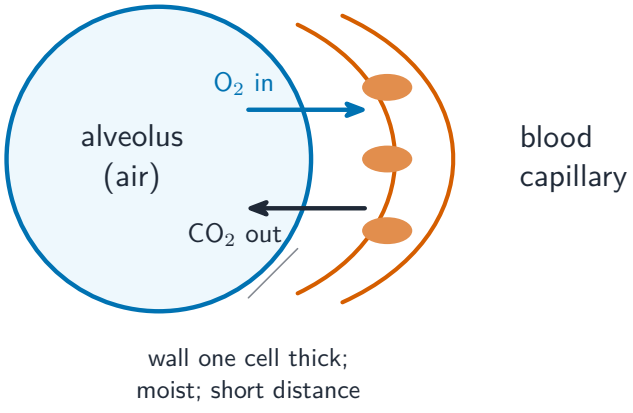
The alveoli are excellent surfaces for exchanging gases, because they have:

- a very large total surface area (millions of tiny sacs),
- very thin walls – the squamous epithelium of the alveolus and the capillary wall are each only one cell thick, so the distance to cross is tiny,
- a rich blood supply from the capillary network,
- a moist lining, so gases dissolve before crossing.

Gases move by **diffusion** down their **concentration gradients** :

- **oxygen** is at a high concentration in the alveolar air and a low concentration in the blood, so it diffuses from the air into the blood.

- **carbon dioxide** is at a high concentration in the blood and a low concentration in the alveolar air, so it diffuses from the blood into the air to be breathed out.



Across the thin, moist wall, oxygen diffuses into the blood and carbon dioxide diffuses out

Under the microscope, real lung tissue looks like a fine pink lace. The many open spaces are the alveoli, and the thin pink lines between them are the walls where gas exchange happens:



Real lung tissue stained for the microscope: the open spaces are alveoli and the thin pink walls are where gases cross; a small airway sits in the centre

Breathing keeps fresh air in the alveoli, and the flowing blood keeps carrying gases away. Both of these keep the concentration gradients steep, so gas exchange stays fast.

Worked example. Use Fick's law to explain why the alveoli allow such rapid gas exchange. Fick's law makes the rate of diffusion proportional to

$$\frac{\text{surface area} \times \text{concentration difference}}{\text{diffusion distance}}$$

so a fast rate needs all three terms working for it. **Surface area**: millions of alveoli give a huge total area, roughly 70 m^2. **Diffusion distance**: the alveolar epithelium and the capillary endothelium are each one flattened cell thick, so oxygen crosses in under 1 μm. **Concentration difference**: ventilation constantly refreshes the air while the circulation constantly carries oxygenated blood away, so a **steep gradient is maintained**. Tie each adaptation to the **term of the equation** it serves - listing "big surface area, thin walls, good blood supply" without linking them to Fick's law is the weaker answer.

Exam tips

- Link each **alveolar** feature to fast diffusion: large surface area, thin (one-cell) walls, moist surface, good blood supply – a "surface area, short distance, steep gradient" answer.
- Match airway tissue to function: cartilage (holds airways open), ciliated + goblet cells (trap and sweep mucus), smooth muscle and elastic fibres.
- Frame answers with **Fick's law** ideas: rate $\propto$ surface area $\times$ concentration difference $\div$ distance.

Infectious diseases

A-Level Biology

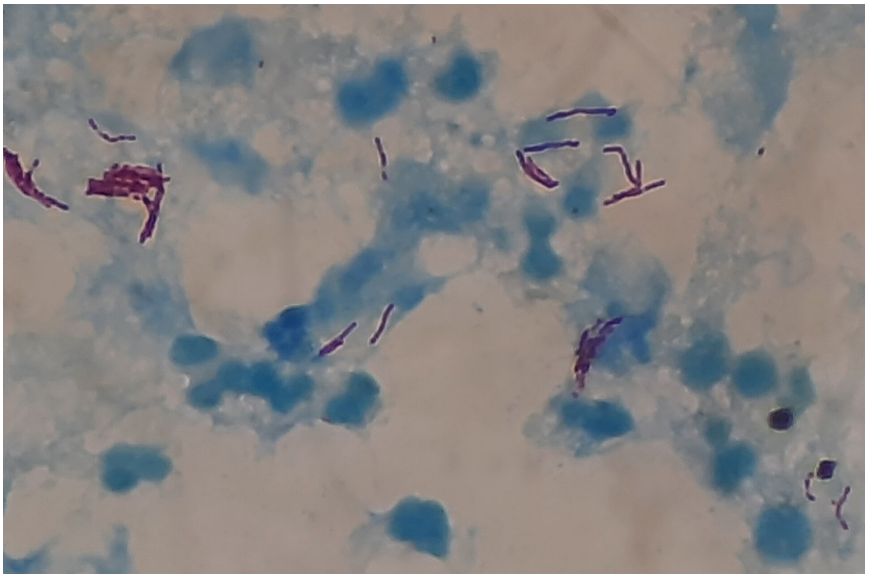
What causes infectious disease

An **infectious disease** is caused by a **pathogen** – an organism that lives in or on a host and causes harm. Infectious diseases are **transmissible**: the pathogen can be **transmitted** (passed) from one person to another.

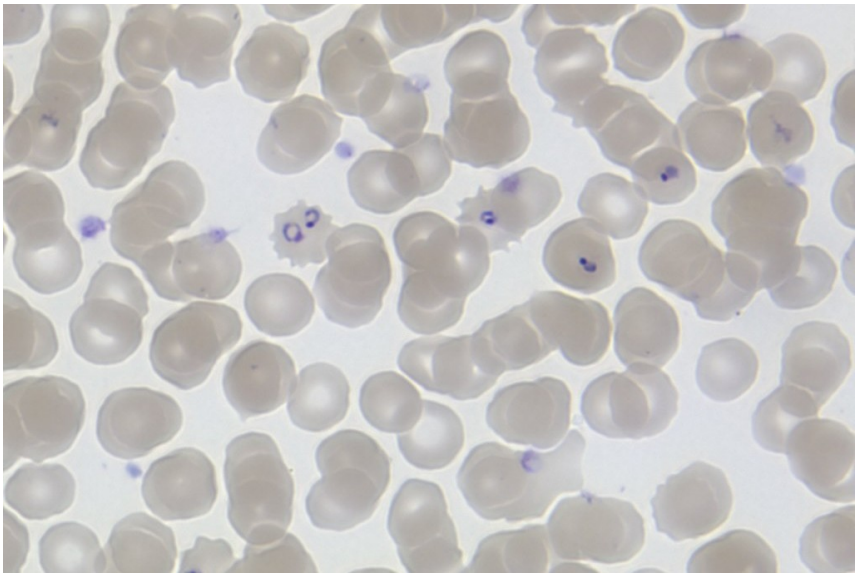
You need to know four diseases, the pathogen that causes each, and how each spreads.

Disease	Pathogen and type	How it spreads
cholera	the **bacterium** *Vibrio cholerae*	drinking water or food **contaminated** with **faeces** (human waste)
malaria	the **protoctist** *Plasmodium*	the bite of an infected **mosquito**, which acts as a **vector** (a carrier of the pathogen); also through infected blood
tuberculosis (TB)	the bacterium *Mycobacterium*	tiny airborne **droplets** from coughs and sneezes; spreads fast where people are crowded
HIV/AIDS	the **virus** HIV (which leads to **AIDS**)	unprotected sex, infected blood (for example shared needles), and from mother to baby

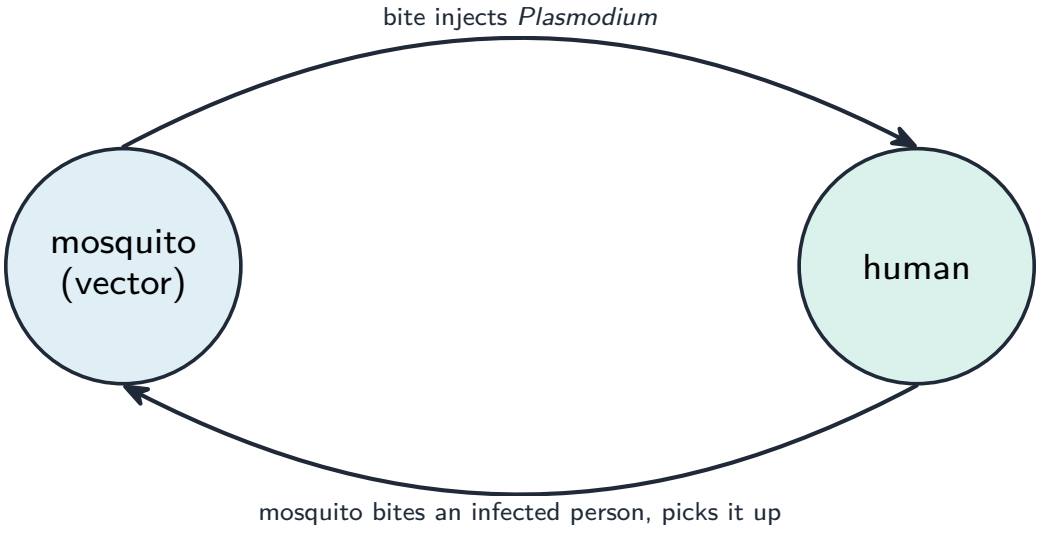
HIV infects and destroys certain white blood cells, so it slowly weakens the body's **immune system**.



The bacterium Mycobacterium (red rods) that causes tuberculosis, stained in a sputum sample



The protoctist Plasmodium (small dark rings) that causes malaria, living inside red blood cells



Malaria needs a mosquito vector: the parasite passes from human to mosquito to human with each bite – which is why control targets the mosquito

Preventing and controlling these diseases

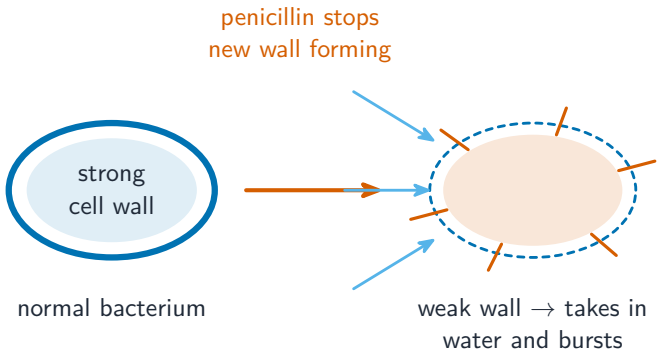
Control has **biological**, **social** and **economic** sides – the science of the pathogen, people's behaviour and education, and the money and resources available. Examples:

- **cholera**: provide clean water and proper sewage treatment; good hygiene; **vaccines** in some areas.
- **malaria**: sleep under nets; remove pools of still water where mosquitoes breed; spray **insecticides**; take anti-malarial drugs.
- **TB**: find and treat infected people with a long course of antibiotics; give the BCG vaccine; reduce overcrowding; trace contacts of patients.
- **HIV**: use condoms; use clean needles; test donated blood; educate people. There is no cure and no vaccine yet, but drugs can slow the virus down.

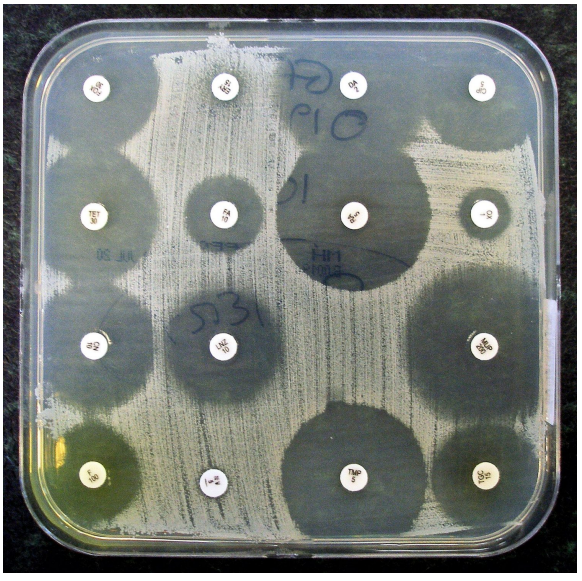
In every case, cost (economic), people's willingness to change behaviour (social) and the supply of drugs or vaccines (biological) all affect how well a disease can be controlled.

Antibiotics

An **antibiotic** is a drug that kills bacteria or stops them growing. For example, **penicillin** stops bacteria from building their **cell walls**. As the bacterium grows, its weak wall cannot hold it, so the cell takes in water and bursts.



Penicillin stops new cell wall forming, so the bacterium takes in water and bursts



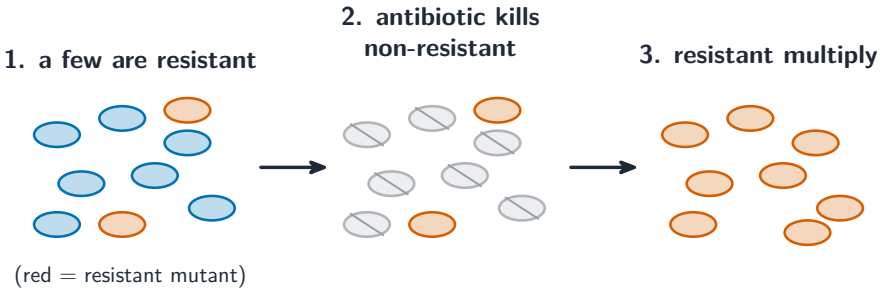
How we test which antibiotic works. Each paper disc holds a different antibiotic. A clear ring means that antibiotic killed the bacteria around it – the bigger the ring, the more effective the drug. Discs with no ring are ones the bacteria resist

Antibiotics do **not** work against **viruses**. A virus has no cell wall and no chemical reactions of its own to attack – it simply uses the machinery of the host cell. So there is no antibiotic target in a virus.

Antibiotic resistance

Sometimes a **mutation** makes a bacterium **resistant** to an antibiotic, which means the antibiotic no longer kills it. This **resistance** is a serious problem:

- when an antibiotic is used, the non-resistant bacteria die, but any resistant ones survive and multiply. Over time, more and more bacteria carry the resistance.
- some infections then become very hard, or impossible, to treat.



Antibiotic resistance spreads by natural selection: the antibiotic kills the rest, so the resistant survivors take over

Steps to slow resistance down:

- only use antibiotics when they are really needed (not for viral illnesses such as colds).
- always finish the full course, so no bacteria are left alive.
- use the correct antibiotic for the infection.
- reduce the heavy use of antibiotics in farming.

Worked example. A patient with influenza is prescribed an antibiotic. Explain why it will not help, and why it is actively harmful. Antibiotics work by attacking structures or processes that **bacteria** have and human cells do not - penicillin, for instance, blocks **cell wall** synthesis, so the growing bacterium bursts under osmotic pressure. A **virus** has no cell wall, no ribosomes of its own and no metabolism: it replicates **inside** the host's cells using the host's machinery, so there is simply no bacterial target for the drug to attack. Taking it anyway exposes the patient's harmless resident bacteria to the antibiotic, killing the susceptible ones and **selecting for resistant** ones - which is how resistance spreads through a population. Name the **target that is missing**: "viruses are not alive" is not the reason and earns nothing.

Exam tips

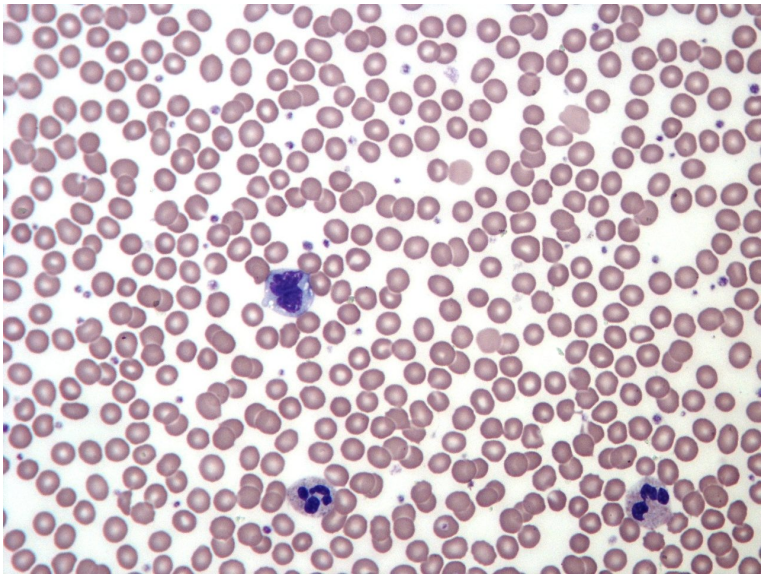
- For each named disease give the **pathogen and route** (cholera – bacterium, water; malaria – Plasmodium, mosquito vector; also TB, HIV/AIDS).
- Explain why **antibiotics** work on bacteria only (they target cell walls/enzymes, not viruses).
- Explain **antibiotic resistance** as natural selection: a resistant mutant survives and reproduces; reduce it by finishing courses and avoiding overuse.

Immunity

A-Level Biology

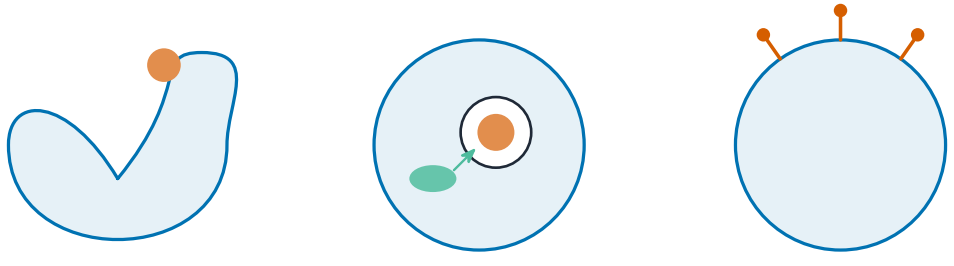
Phagocytes – the first defence

Your **immune system** is the set of cells that defends the body against **pathogens**. The first cells to act are **phagocytes**, a group of white blood cells that includes **macrophages** and **neutrophils**.



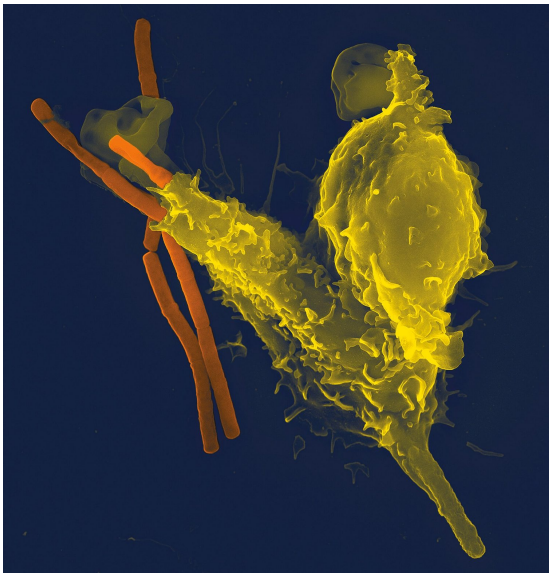
A blood smear: the large stained cells are white blood cells, scattered among the many red cells

A phagocyte destroys a pathogen by **phagocytosis**: it surrounds the pathogen, takes it inside in a vesicle, and digests it with enzymes. After this, a macrophage displays parts of the pathogen on its own surface, ready to alert other immune cells.



Phagocytosis: the phagocyte engulfs the pathogen, digests it, then displays its antigen

This false-coloured electron micrograph captures the moment in real life: a phagocyte reaching out to grab and engulf rod-shaped bacteria.

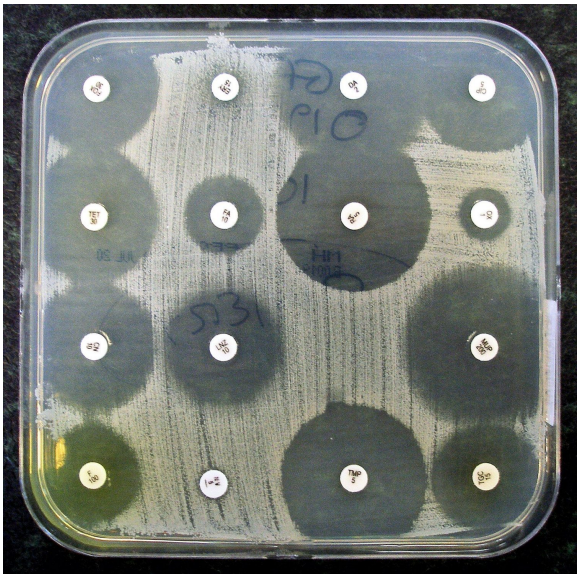


*A real **neutrophil** (yellow), one kind of phagocyte, reaching out to engulf rod-shaped bacteria (orange) by phagocytosis*

Antigens: self and non-self

An **antigen** is a molecule (usually a protein) on a cell surface that the immune system can recognise.

- **self antigens** are the body's own markers. The immune system learns to ignore them.
- **non-self antigens** are foreign, for example the antigens on a pathogen. These trigger an immune response.

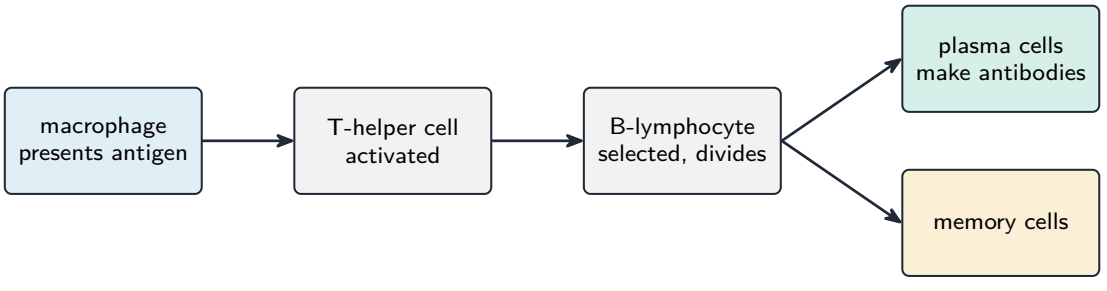


Clear zones on a culture plate: the immune system recognises non-self antigens and attacks foreign cells

The primary immune response

The first time a new pathogen enters, the body makes a slow **immune response** . The main steps are:

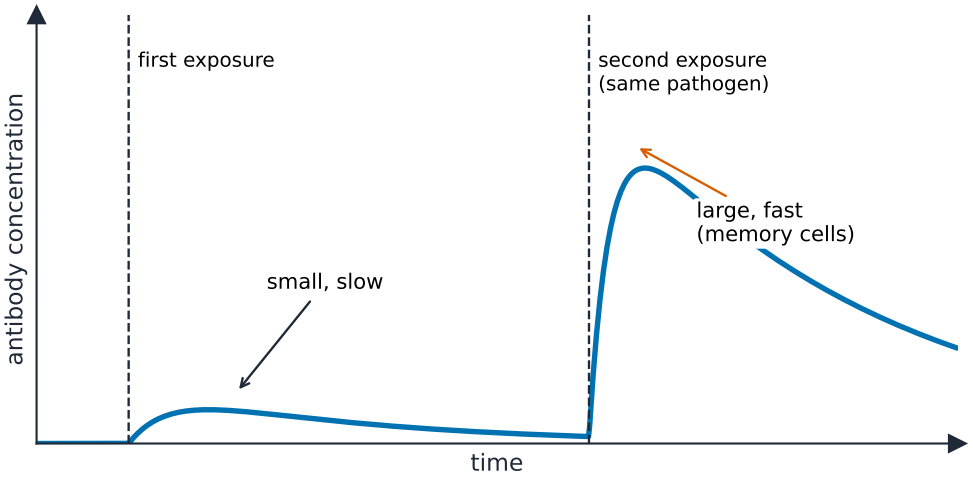
1. a macrophage engulfs the pathogen and displays its antigen.
2. **T-helper cells** T recognise that antigen and become active. They release chemicals that switch on other cells.
3. **B-lymphocytes** with a matching shape are selected. They divide to form **plasma cells** , which pour out **antibodies** , and **memory cells** .
4. **T-killer cells** T destroy the body's own cells that have been infected.



Only the B-lymphocyte whose shape matches the antigen is selected and cloned – into antibody-making plasma cells and long-lived memory cells

Memory cells and long-term immunity

Memory cells stay in the body for years after the infection is over. If the **same** pathogen enters again, the memory cells start a **secondary** immune response that is much faster and larger than the first. The pathogen is destroyed before it can make you ill. This is what we mean by long-term immunity.

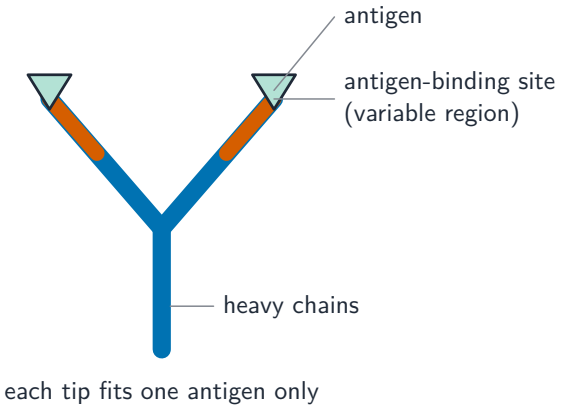


secondary response: faster and larger (memory cells)

The secondary response is faster and larger, because memory cells are ready

Antibodies

An antibody is a Y-shaped protein made by plasma cells. The two tips of the Y are **antigen-binding sites** . Each site has a special shape, called the **variable region** , that fits one antigen only, like a lock and key.



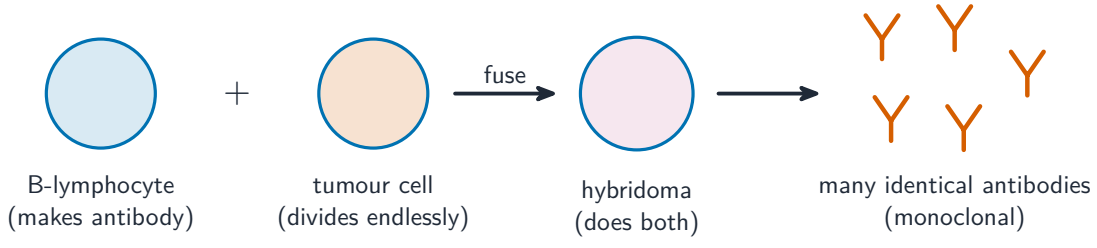
An antibody is Y-shaped; the two tips are the variable regions that bind one specific antigen

Antibodies help in several ways: they stick to antigens, clump pathogens together so they are easier to deal with, mark pathogens so phagocytes find them, and block harmful toxins.

Monoclonal antibodies

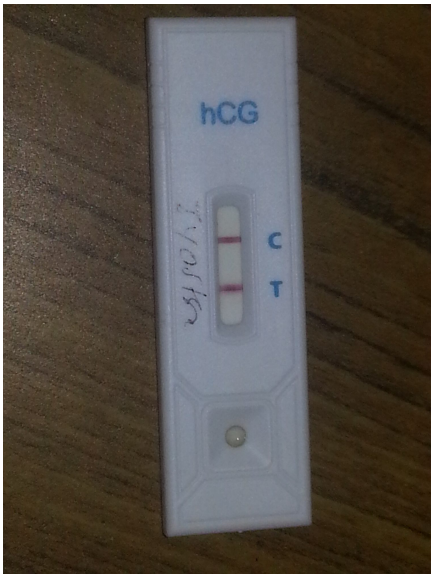
A **monoclonal antibody** is a single type of antibody, all identical. They are made by the **hybridoma** method:

1. an animal is given an antigen, so it makes B-lymphocytes that produce the wanted antibody.
2. these B-lymphocytes are fused with tumour cells, which divide endlessly.
3. the fused cell (the hybridoma) both makes the antibody **and** divides without stopping, producing large amounts of one identical antibody.



Fusing a B-lymphocyte with a tumour cell makes a hybridoma that pours out one identical (monoclonal) antibody

Monoclonal antibodies are used in the **diagnosis** of disease (to detect a specific molecule, as in a pregnancy test) and in treatment (to carry drugs to specific target cells, such as cancer cells).

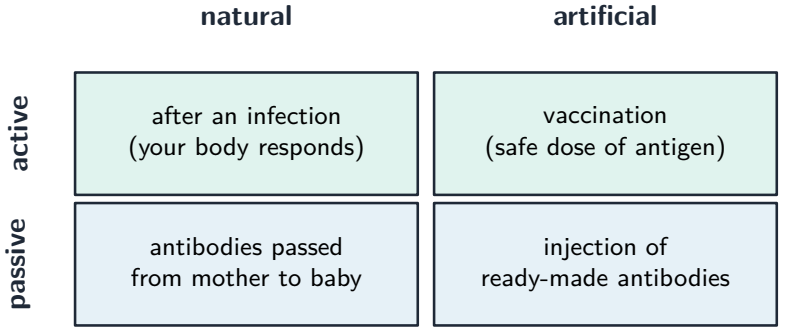


A pregnancy test is monoclonal antibodies at work. The antibodies are fixed at the T line and grab only one molecule – the pregnancy hormone hCG. The T line turns red only if hCG is in the urine; the C line always appears, to show the test ran properly

Types of immunity

Immunity can be active or passive, and natural or artificial.

- **active immunity** – your own body meets an antigen and makes its own antibodies and memory cells. It is slow to start but long-lasting.
- **passive immunity** – ready-made antibodies are given to you from outside. It works at once but does not last, because there are no memory cells.
- **natural immunity** – gained in a natural way (active: after an infection; passive: antibodies passed from mother to baby).
- **artificial immunity** – gained on purpose (active: your body is made to respond to a safe dose of antigen; passive: you are injected with ready-made antibodies).



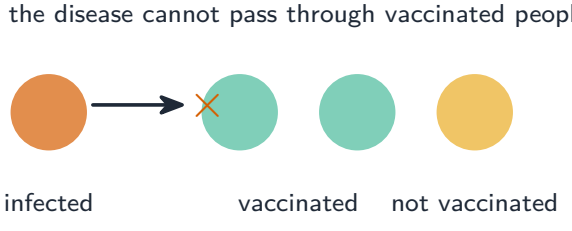
active = your body makes its own antibodies (slow, long-lasting);
passive = ready-made antibodies given (fast, short-lived)

Active immunity (your body responds) lasts; passive immunity (ready-made antibodies) is fast but short

Vaccination

A **vaccine** contains antigens – often a dead or weakened pathogen, or part of one. The antigens trigger a primary immune response and make memory cells, so you gain long-term immunity **without** becoming ill.

Vaccination programmes can control the spread of a disease across a population. If enough people are vaccinated, the pathogen cannot pass easily from person to person. This protects even the people who are not vaccinated, an effect called **herd immunity**.



Herd immunity: when enough people are vaccinated, the pathogen cannot reach the few who are not

Worked example. Classify each as active or passive, natural or artificial: (a) a baby receives antibodies in breast milk; (b) a child is vaccinated against measles; (c) someone recovers from chickenpox; (d) a patient bitten by a snake is given antivenom. Ask two questions each time. **Did the person make the antibodies themselves?** If yes it is **active**; if they were handed ready-made ones, it is **passive**. **Did it happen by chance or deliberately?** Naturally, or artificially. So (a) is **natural passive** - ready-made antibodies by a natural route; (b) is **artificial active** - the vaccine's antigens make the child produce their own; (c) is **natural active** - the infection made them produce their own; (d) is **artificial passive** - ready-made antibodies given deliberately. Only **active** immunity makes **memory cells**, which is exactly why passive immunity acts immediately but is **short-lived**.

Exam tips

- Distinguish **phagocytes** (engulf, non-specific) from lymphocytes: B cells → antibodies (humoral), T cells (helper/killer, cell-mediated).
- Label an **antibody** (variable region, antigen-binding site) and explain agglutination and neutralisation.
- Compare the **primary and secondary response** on a graph – memory cells make the secondary faster and larger.
- Distinguish **active vs passive** and **natural vs artificial** immunity with an example of each; explain vaccination and herd immunity.

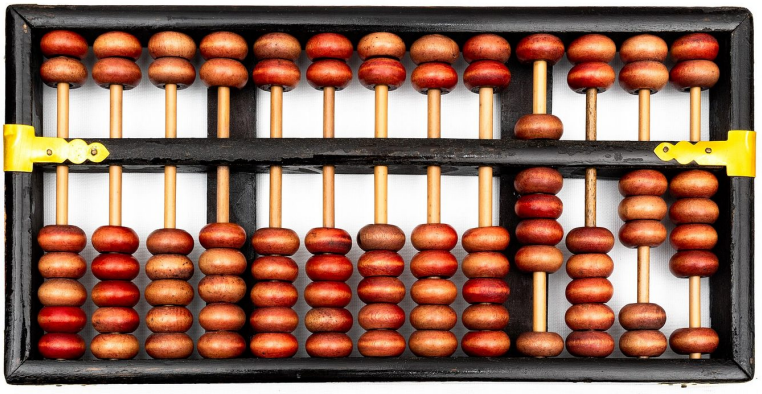
Information representation

A-Level Computer Science

Number systems

The three **number systems** you must use:

- **denary** (decimal, base 10) – uses digits 0–9. Place values are powers of ten.
- **binary** (base 2) – uses 0 and 1. Place values are powers of two. Every **byte** is 8 **bits**.
- **hexadecimal** (base 16) – uses 0–9 then A–F for 10–15. Each hex **digit** stands for exactly 4 bits.



An abacus represents numbers by place value – the same idea behind decimal, binary and hexadecimal

Conversions

Denary → binary: keep dividing by 2 and record the remainders, read bottom-up. Or subtract the largest **place value** (power of 2) that fits.

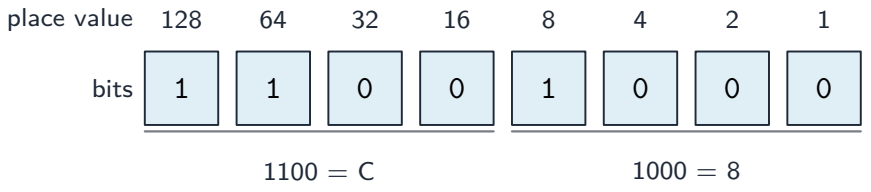
Example: 558_{10}: $558 = 512 + 32 + 8 + 4 + 2 = 2^9 + 2^5 + 2^3 + 2^2 + 2^1$. In 12 bits: `0010 0010 1110`.

Binary → hex: group the bits into **nibbles** (4 bits) from the right and convert each. `0010 0010 1110` → `2 2 E` → `22E`.

Hex → binary: replace each hex digit with its 4-bit pattern. **Hex → denary**: multiply each digit by its place value. `22E` $= 2 \times 256 + 2 \times 16 + 14 = 558$.

Worked example. Convert denary 200 to 8-bit binary, then to hexadecimal.

$200 = 128 + 64 + 8$, so the binary is `11001000`. In nibbles, `1100 1000` = 12 and 8, i.e. C and 8, so the hexadecimal is `C8`.



$$200 = 128 + 64 + 8 = \texttt{11001000}_2 = \texttt{C8}_{16}$$

Reading 200 from its place values, then grouping the bits into nibbles to get hex C8

How many bits?

Exam questions fix the **register width** (8, 12 or 16 bits). Pad with leading zeros to that width: 558 in 12 bits is `0010 0010 1110`, never `10 0010 1110`.

To find the **minimum number of bits** that can store a value, ask which place values you need:

- an unsigned integer from 0 to $2^n - 1$ needs n bits: 200 needs 8 bits (the top is 255), 1000 needs 10 bits (the top is 1023), 16 needs 5 bits (4 bits stop at 15).
- a signed two's-complement integer from -2^{n-1} to $2^{n-1} - 1$ needs n bits: -200 needs 9 bits, because 8 bits stop at -128.
- one hexadecimal digit needs 4 bits, one BCD digit needs 4 bits, and one ASCII character needs 7 bits (8 for extended ASCII).

Binary vs decimal prefixes

Two prefix families look similar but differ – decimal (powers of 10) and binary (powers of 2):

Decimal (SI)	Binary (memory)
kilo = 10^3	kibi (Ki) = $2^{10} = 1024$
mega = 10^6	mebi (Mi) = 2^{20}
giga = 10^9	gibi (Gi) = 2^{30}
tera = 10^{12}	tebi (Ti) = 2^{40}

So a tebibyte (TiB) is slightly more than a terabyte (TB). A "1 TB" drive holds 10^{12} bytes, but an operating system that reports in TiB shows a smaller number.

Binary arithmetic

■ Binary addition

Add column by column from the right, carrying as in denary:

Bit A	Bit B	Carry in	Sum bit	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	1	0	0	1
1	1	1	1	1

Overflow happens when the result needs more bits than the **register** can hold – the carry-out of the leftmost column is the overflow bit.

> **Worked example.** Add the 8-bit unsigned integers 10110101 and 01101100, and comment on the result.

10110101 + 01101100 = 1 00100001. The answer needs 9 bits, so it does not fit in an 8-bit register: overflow has occurred. A full answer names the error and says why, using the word size the question gave: "Overflow: the true result (289) is larger than the largest value an 8-bit register can hold (255), so the carry out of the most significant bit is lost and the stored result (00100001 = 33) is wrong."

■ Binary subtraction

The usual way is **two's complement** addition: to do $A - B$, form the two's complement of B (invert every bit and add 1), then add, and discard any final carry-out.

To subtract 00011110 from 01100100 (unsigned 8-bit):

- two's complement of 00011110: invert → 11100001, add 1 → 11100010.
- add to 01100100: result 1 01000110 (9 bits) – discard the leading 1 → 01000110 = 70_{10}. Check: $100 - 30 = 70$. ✓

■ Two's complement signed integers

In an n-bit two's-complement number:

- the **most significant bit** (MSB) is the **sign bit** : 0 = positive, 1 = negative.
- to read a negative number: invert every bit, add 1, then negate.

So 11100010 is negative; invert → 00011101, add 1 → 00011110 = 30, so it is −30. This is a **signed integer** (unlike an **unsigned** one). The range for n bits is -2^{n-1} to $+2^{n-1} - 1$; for 8 bits, −128 (10000000) to +127 (01111111).

The same bits mean different numbers depending on the agreed reading. As an **un-**

signed integer every bit is a place value, so 8 bits run from 0 to 255; as a **signed** two's-complement integer the top bit is the sign, so the same 8 bits run from −128 to +127. The pattern 11111111 is 255 read one way and −1 read the other – nothing in the bits themselves says which.

8 bits	as unsigned every bit a place value	as signed top bit is the sign
00000000	0	0
01111111	127	+127
10000000	128	−128
11111111	255	−1
range for n bits	$0 \ldots 2^n - 1$	$-2^{n-1} \ldots 2^{n-1} - 1$

The same byte read as unsigned and as signed: only the agreed interpretation tells them apart  *8-bit two's complement: the sign bit splits the range into negative (−128 to −1) and positive (0 to 127)*

> **Worked example.** What denary value does the 8-bit two's-complement number 10110100 represent?

The MSB is 1, so it is negative. Invert → 01001011, add 1 → 01001100 = 76, so the value is −76. Check with place values: $-128 + 32 + 16 + 4 = -76$.

> **Worked example.** Write −108 as a 12-bit two's-complement integer.

Start from +108 in 12 bits: $108 = 64 + 32 + 8 + 4$, so `0000 0110 1100`. Invert every bit: `1111 1001 0011`. Add 1: `1111 1001 0100`. Check with place values, where the top bit is worth $-2^{11} = -2048$: $-2048 + 1024 + 512 + 256 + 128 + 16 + 4 = -108$. ✓

For 12 bits the range is −2048 (`1000 0000 0000`) to +2047 (`0111 1111 1111`). Questions that ask for the smallest and largest values want these two patterns, so learn the rule: the most negative number is a 1 followed by zeros; the most positive is a 0 followed by ones.

An **arithmetic shift** moves every bit left or right but keeps the sign: a shift right by one place halves the value and copies the sign bit into the empty space on the left, so a negative number stays negative (`1111 1001 0100` shifted right three places is `1111 1111 0010`, which is −14: $-108/8 = -13.5$, and a shift right rounds down). A shift left doubles the value. Shifts belong to the assembly instruction set in topic 4, but this question is asked with the number work here.

Overflow in signed arithmetic happens when the true result falls outside this range – spotted when the sign bit flips wrongly (two positives giving a negative, or two

negatives giving a positive).

■ One's complement

Before two's complement, an older scheme called **one's complement** represented a negative number by simply **inverting every bit** of the positive – there is no "add 1" step.

- $+30 = 00011110$, so in one's complement $-30 = 11100001$ (just the inverse).
- Drawback: it has **two zeros** – 00000000 ($+0$) and 11111111 (-0) – which wastes a bit pattern and makes arithmetic awkward.

Two's complement (invert **and** add 1) removes the negative zero: it has a single zero and lets addition and subtraction use the same circuit. That is why modern computers store signed integers in two's complement, not one's complement.

Binary Coded Decimal (BCD)

In **BCD** , each denary digit is written as its own 4-bit pattern. The number 93 is `1001 0011` in BCD – not binary 93 (01011101). Each nibble uses only 0–9; patterns 1010–1111 are invalid.

BCD reading: `0010 0111 0101` → 2, 7, 5 → 275.

Use: calculators, digital clocks, and devices that show denary digits – each digit drives a **7-segment display** . Currency code often uses BCD to avoid the rounding errors of converting fractions like 0.1 to binary.

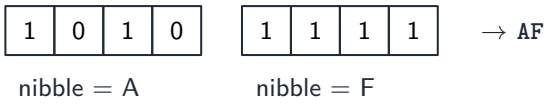
A "justify" answer must link the use to a property of BCD: each denary digit has its own 4 bits, so a digit can be sent straight to its display, or added digit by digit, with no conversion of the whole number; and a decimal fraction such as 0.10 is stored exactly, which a binary fraction cannot do.



A seven-segment display shows one denary digit, often driven by BCD

Hexadecimal – practical uses

Hex is a compact way to write binary (1 hex digit = 4 bits):



A byte is two nibbles; each nibble is one hex digit

- **memory addresses** in low-level programming – `0x7FFE`.
- **colour values** in HTML/CSS – `#FF8800`.
- **MAC addresses** – `AC:DE:48:00:11:22`.

Hex does not change the stored data – it just makes binary easier for humans.

Character codes

Computers store text as numbers; each character has a numeric **code point** set by a **character set** .

ASCII

- **ASCII** uses 7 bits – 128 code points. Basic Latin letters, digits, punctuation, and control codes.
- **Extended ASCII** uses 8 bits – 256 code points; the lower 128 match ASCII, the upper 128 vary by region.

character	code (denary)	binary
`A`	65	`01000001`
`a`	97	`01100001`
`0`	48	`00110000`
`(space)`	32	`00100000`

Each character is stored as a number – a few ASCII code points in denary and binary

Unicode

- **Unicode** is a universal character set covering almost every script, plus symbols and emoji.
- common **encodings** : **UTF-8** (1–4 bytes, ASCII-compatible), **UTF-16** (2 or 4 bytes), **UTF-32** (fixed 4 bytes).

Why Unicode beats ASCII

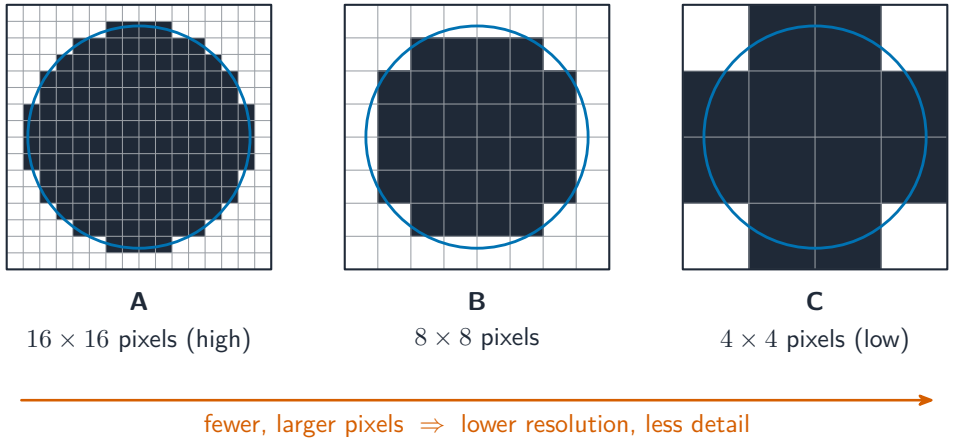
- it represents **far more characters** (every script, emoji); ASCII covers only basic English.
- files are **portable** with no code-page confusion, and allow **multilingual** text in one document.
- trade-off: Unicode files are usually **larger** for English-only text.

When a question asks for **differences**, give them in pairs with numbers: ASCII uses 7 bits (extended ASCII 8), so 128 (256) characters; Unicode uses up to 32 bits (UTF-8 uses 1 to 4 bytes), so more than a million code points. ASCII covers basic English only; Unicode covers every script, and its first 128 code points are the ASCII ones. In UTF-8 an English letter still takes 1 byte, so a 40-letter English file name is 40 bytes in ASCII and in UTF-8 alike, while a Chinese character takes 3 bytes.

Bitmap images

A **bitmap** image (also called a **bitmapped image**) stores the colour of every **pixel** in a grid. At the start of the file a **file header** records the image's metadata – its width, height and colour depth – so software knows how to read the pixel data that follows.

- **image resolution** : the bitmap's own size, width × height in pixels (e.g. 1920 × 1080).
- **screen resolution** : the width × height the **display** can show. If an image's resolution is larger than the screen it is scaled down to fit; a low-resolution image looks blocky when stretched onto a higher-resolution screen.
- **colour depth** (**bit depth**): bits per pixel. 1 bit → black/white; 8 bits → 256 colours; 24 bits → 16.7 million ("true colour").

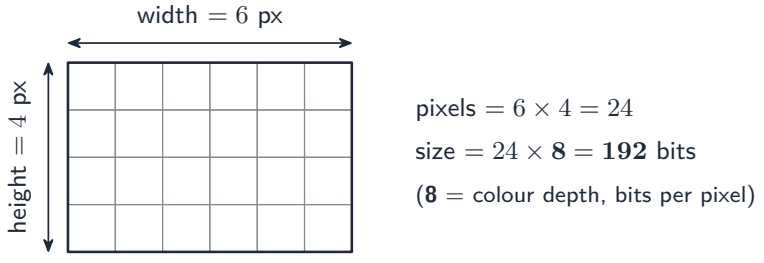


The same image stored at three resolutions, from high (A) to low (C): fewer, larger pixels mean less detail

File size

$$\text{size in bits} = \text{width} \times \text{height} \times \text{bit depth}.$$

Divide by 8 for bytes, by 1024 for KiB, etc. Example: a 3000×2000 image at 24 bpp is $3000 \times 2000 \times 24 = 1.44 \times 10^8$ bits ≈ 17.2 MiB.



The same formula on small numbers: count the pixels, then multiply by the colour depth

State the units you used. The mark scheme accepts 1 MB $= 10^6$ bytes (the SI prefix) or 1 MiB $= 1024 \times 1024$ bytes (the binary prefix), as long as your working shows which one; the same image is 18.0 MB or 17.2 MiB. Add the size of the file header if the question gives one.

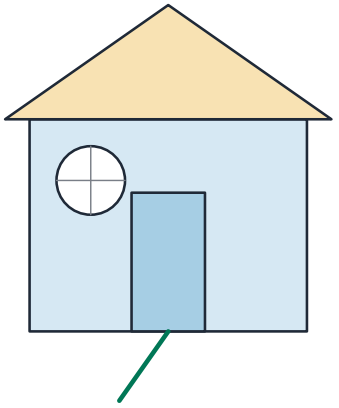
A video is a sequence of bitmap images, each one a **frame** . Before compression its size is the size of one frame $\times$ the **frame rate** (frames per second) $\times$ the duration in seconds: 30 frames per second of 1920×1080 pixels at 24 bits is $30 \times 1920 \times 1080 \times 24 \approx 1.5 \times 10^9$ bits, about 187 MB, for every second. That is why video is always compressed.

Changing settings

- **lower resolution** → smaller file, less detail (looks blocky when enlarged).
- **lower colour depth** → smaller file, but smooth shades show banding.
- **higher** of either → larger file, better quality.

Vector graphics

A **vector graphic** stores the **instructions** to draw the image as a **drawing list** – an ordered list of **drawing objects** (geometric **primitives** : lines, curves, polygons, circles). Each drawing object has **properties** such as colour, fill, line width and position (coordinates). To show it, the program **renders** the drawing list at any resolution needed.



the file is a list of shapes:

triangle — roof	3 points
circle — window	$(0.75, 1.85)$, $r\,0.42$
rectangle — body	$(0, 0)$, 3.4×2.6
rectangle — door	$(1.25, 0)$, 0.9×1.7
line — path	$(1.7, 0) \to (1.1, -0.85)$

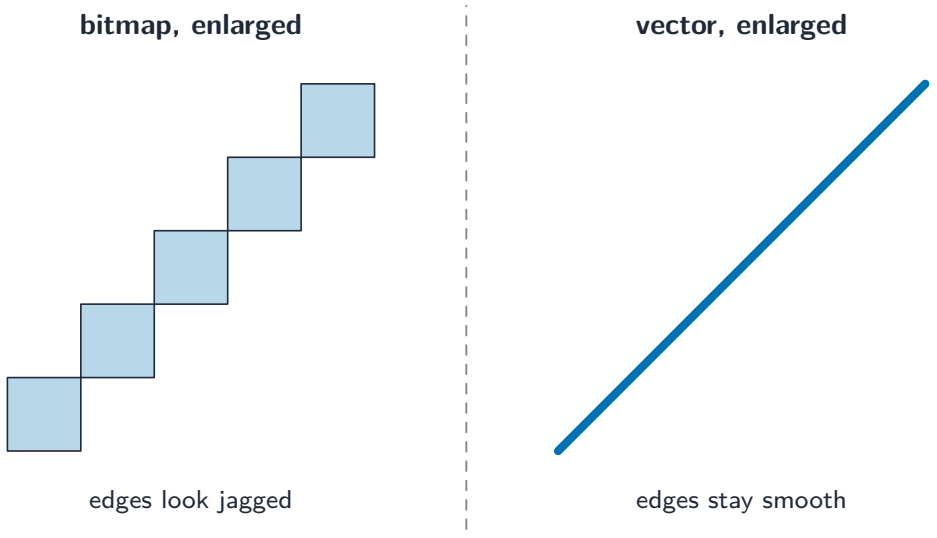
A vector image is built from labelled geometric shapes, each with attributes

Bitmap vs vector

Task	Better choice	Why
Photograph	Bitmap	Complex pixel-level detail can't be described as shapes.
Logo, icon, sign	Vector	Sharp edges; scales to any size without blur.
Engineering drawing	Vector	Precise geometry and scaling.
Painting, texture	Bitmap	Smooth tonal detail per area.

Vector advantage: it **scales without losing quality** – a vector logo stays sharp at any size, while a bitmap blurs when enlarged. **Vector disadvantage**: it cannot describe arbitrary pixel detail (photographs).

A "justify" answer links the choice to the task. "The logo must appear on a business card and on a billboard, so it should be a vector graphic: it is stored as drawing objects and is re-rendered sharply at any size, whereas a bitmap would show its pixels when enlarged." For a photograph the argument runs the other way: there are no shapes to describe, so every pixel's colour must be stored.

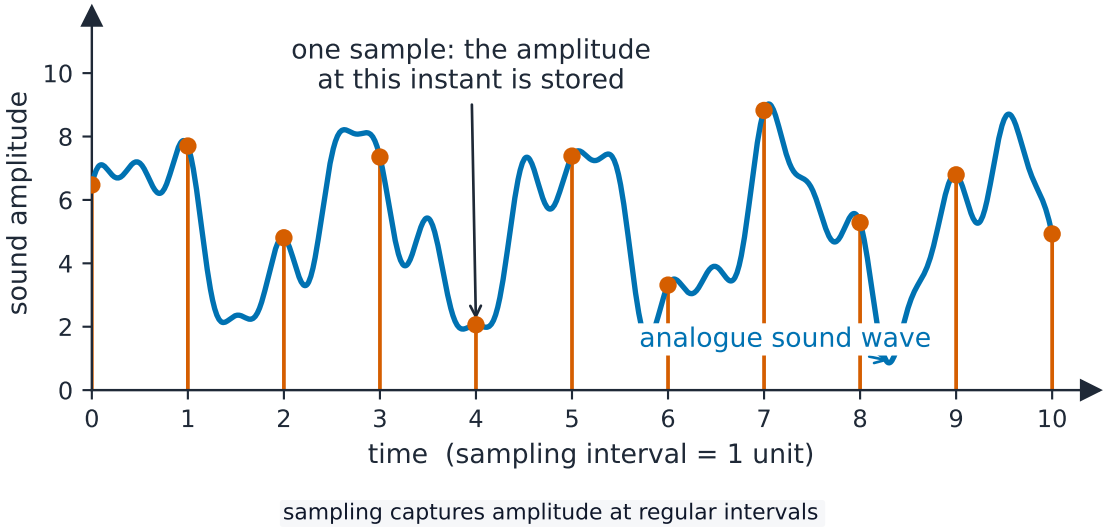


Enlarged, a bitmap's pixels turn jagged; a vector stays smooth at any size

Sound

A continuous wave of **analogue data** (the sound) is converted into **digital data** by **sampling** :

- **sampling rate** – samples per second (Hz). CD quality is 44.1 kHz.
- **sampling resolution** (bit depth) – bits per sample's **amplitude** . CD quality is 16 bits.



Sampling a sound wave: its amplitude is read at each time interval

File size

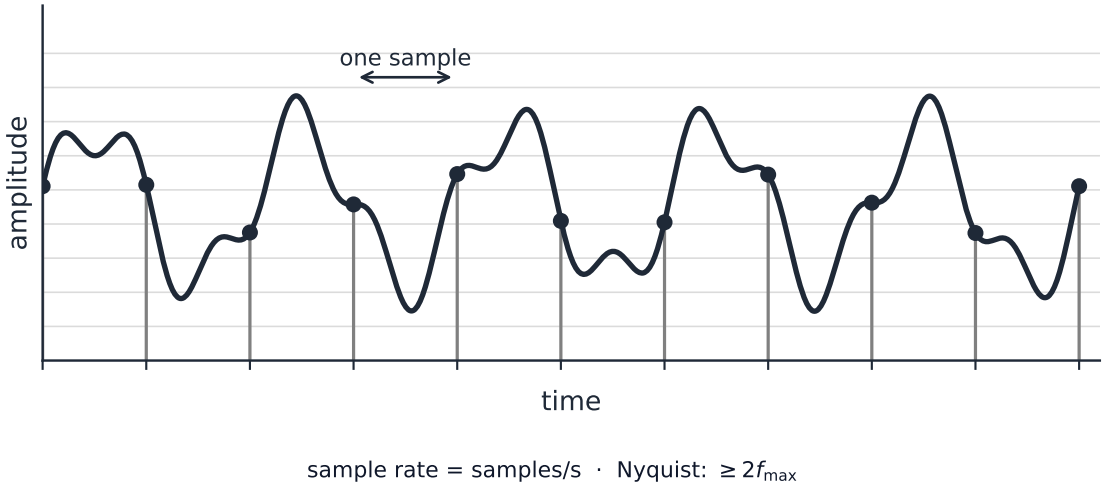
$$\text{size in bits} = \text{sampling rate} \times \text{resolution} \times \text{duration} \times \text{channels}.$$

A 10-second stereo CD clip: $44100 \times 16 \times 10 \times 2 = 14\,112\,000$ bits ≈ 1.68 MiB.

Changing settings

- **higher sampling rate** → captures higher pitches, larger file.
- **higher sample resolution** → finer amplitude steps, less **quantisation** noise, larger file.
- **lower** of either → smaller file, clear quality loss.

(The sampling rate must be at least twice the highest frequency you want to keep.)



Sample rate is samples per second; the Nyquist rule is why it must be at least twice the highest frequency kept

Compression

Compression reduces file size, saving storage and transmission **bandwidth** . Two kinds:

- **lossless** – the original data is recovered exactly (text, programs, ZIP/PNG).
- **lossy** – some detail is dropped for much smaller files (JPEG, MP3, video).

When to use which

- **lossless** for documents, source code, medical images – anything needing exact data.
- **lossy** for streaming media. **Real-time video streaming** uses lossy compression because it must send huge amounts of data in real time over limited bandwidth; lossless would not shrink it enough. Raw HD video is gigabytes per minute, so without compression the picture would keep freezing.

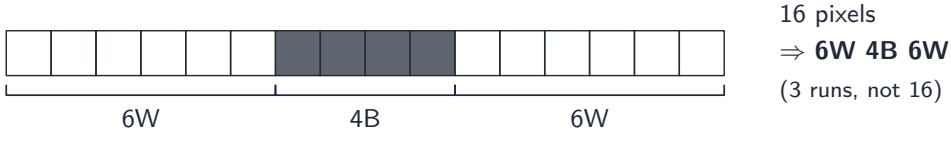
A "justify" answer names the method, then the reason from the situation: "Lossless, because the spreadsheet must be restored exactly; a single changed value would make the accounts wrong." Or: "Lossy, because the photographs are viewed on a phone screen where the dropped detail is not visible, and the smaller files upload faster and use less storage."

Lossless methods

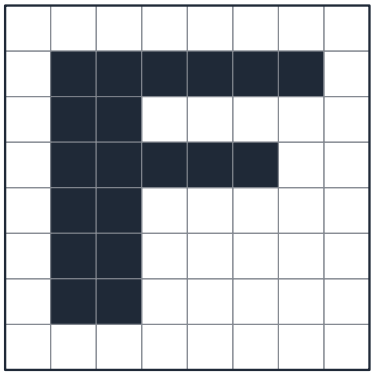
- **run-length encoding** (RLE): store "the next n values are x" instead of repeating x. Great for flat areas; useless for noisy data.
- **dictionary methods** (ZIP, PNG): replace repeated byte sequences with a short reference. Good for text and code.
- **Huffman coding** : give short codes to common symbols and long codes to rare ones, bringing the average code length near the data's **entropy** .

How each kind of file is compressed:

- **text file**: dictionary methods and Huffman coding turn repeated words and common characters into short codes. Text must stay lossless, because one changed character changes the meaning.
- **bitmap image**: RLE for runs of identical pixels (icons, diagrams, black-and-white scans); lossy JPEG for photographs, or a lower colour depth or resolution.
- **vector graphic**: the drawing list is already small; remove drawing objects that are not needed, store coordinates to fewer decimal places, or apply a lossless method such as ZIP to the file.
- **sound file**: lossy MP3 or AAC removes what the ear cannot hear; a lower sampling rate or resolution is also lossy; lossless formats keep every sample and shrink the file much less.

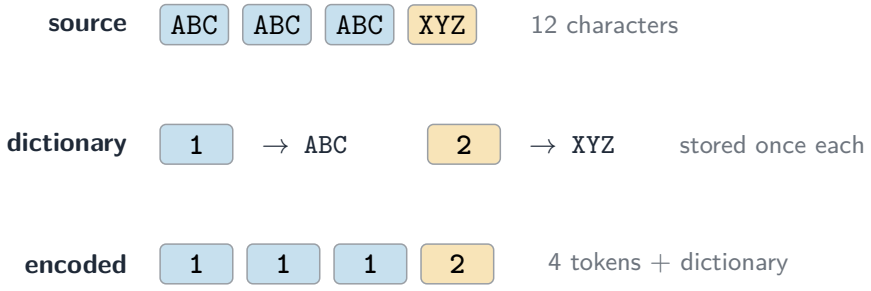


Run-length encoding on a single row: 16 pixels become 3 runs

8×8 bitmap	binary	RLE code
	00000000	8W
	01111110	1W 6B 1W
	01100000	1W 2B 5W
	01111100	1W 5B 2W
	01100000	1W 2B 5W
	01100000	1W 2B 5W
	01100000	1W 2B 5W
	00000000	8W

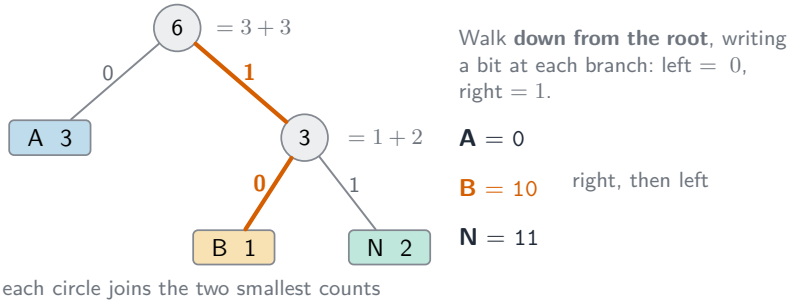
Key: 1 = black (B), 0 = white (W). RLE stores *count* + *colour*: e.g. 01111110 → 1W 6B 1W

Run-length encoding of the letter F in an 8 × 8 black-and-white grid



The decoder rebuilds the dictionary as it reads, so the original returns exactly — **lossless** (ZIP, PNG).

Dictionary coding: each repeated sequence is stored once, and every occurrence becomes a short index

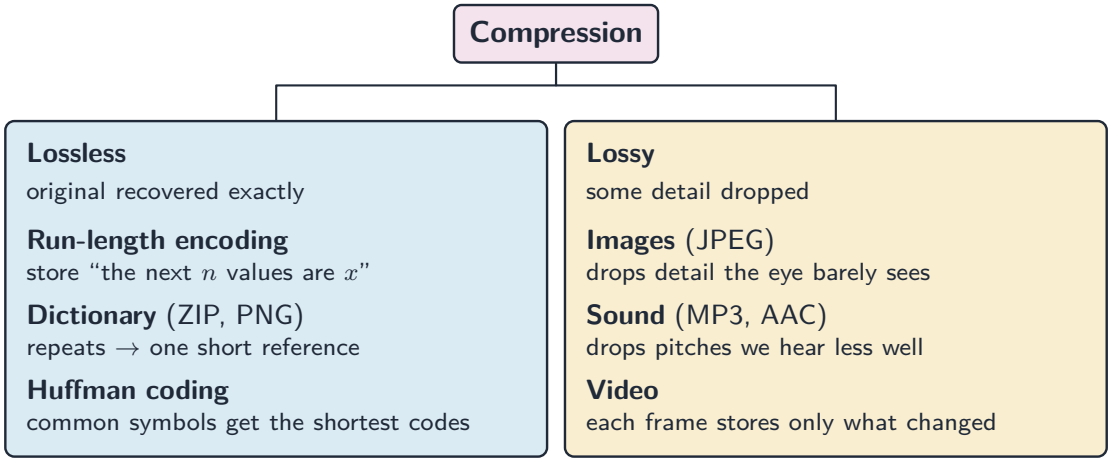


BANANA becomes 10 0 11 0 11 0 — **10 bits**, against 12 for fixed 2-bit codes: the commonest letter is nearest the root.

Huffman coding: the commonest symbol gets the shortest code, so BANANA needs 10 bits instead of 12

Lossy methods

- **images** (JPEG): drop fine detail and colour differences the eye barely sees.
- **sound** (MP3, AAC): drop pitches we hear less well, and quiet sounds hidden by louder ones.
- **video** combines **spatial** compression (within each frame, like JPEG) with **temporal** compression (most frames store only the differences from the previous frame).



Compression methods: lossless versus lossy, with common examples

Logical and arithmetic shifts; IP address sizes

A **logical shift** moves the bit pattern and fills vacated positions with 0. In a **right arithmetic shift** of a two's-complement integer, the sign bit is copied into the new leftmost positions. For eight-bit `11110100` (−12), logical right by two gives `00111101` (61); arithmetic right by two gives `11111101` (−3). Discarded bits are lost; overflow and rounding can prevent shifts behaving like exact multiplication or division.

An **IPv4 address** has 32 bits, often shown as four eight-bit decimal groups. An **IPv6 address** has 128 bits, usually shown as eight groups of four hexadecimal digits before permitted abbreviation. These are address lengths, not numbers of characters in the printed address.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
bit	a single binary digit, 0 or 1
byte	a group of 8 bits
binary prefix	a multiplier that is a power of 2 (kibi = 1024) rather than a power of 10 (kilo = 1000)
two's complement	a way of representing signed integers in which the most significant bit has a negative place value
overflow	the result of a calculation is too large to be represented in the number of bits available
Binary Coded Decimal	each denary digit is stored as its own 4-bit binary pattern
character set	the set of characters a computer can represent, each with its own binary code
pixel	the smallest element of a bitmap image, storing one colour value
image resolution	the number of pixels in an image, given as width by height
screen resolution	the number of pixels a display can show, given as width by height
colour depth	the number of bits used to store the colour of one pixel
sampling rate	the number of samples of the sound taken per second

Term	Definition
sampling resolution	the number of bits used to store the amplitude of one sample
lossless compression	compression from which the original data can be recovered exactly
lossy compression	compression that permanently removes some data, so the original cannot be recovered
run-length encoding	replacing a run of repeated values with one value and a count

Exam tips

- Show working for base conversions: denary → binary by place values, binary → hexadecimal in **nibbles** (groups of 4 bits).
- For **two's complement** the MSB is negative; to negate, **invert and add 1**; watch for **overflow** when the sign bit flips wrongly.
- Distinguish **bitmap** (pixels; file size = width × height × colour depth) from **vector** (drawing commands; scales without loss).
- Sound file size depends on **sample rate** × **bit depth** × **time** – more of each means better quality but a bigger file.
- Compare **lossless vs lossy** compression and give a use for each.

Common mistakes

- Explaining an overflow with "the answer was greater than 255" or "it has 9 bits". State the word size the question gave, then say the result cannot be represented in it.
- Making a negative number by setting the top bit to 1 and leaving the rest (sign and magnitude). Two's complement means invert every bit of the positive value, then add 1.
- Forgetting to pad a converted number to the register width the question asks for.
- Mixing bits and bytes in a file-size calculation. Work in bits, divide by 8 once, and say whether you used 1000 or 1024.
- Answering "describe" in everyday words ("the picture gets worse"). Use the syllabus terms: fewer colours, banding, lower image resolution, larger pixels.

Communication

A-Level Computer Science

Networks: purpose and benefits

A **network** is a set of computing devices connected so they can communicate and share resources. Benefits:

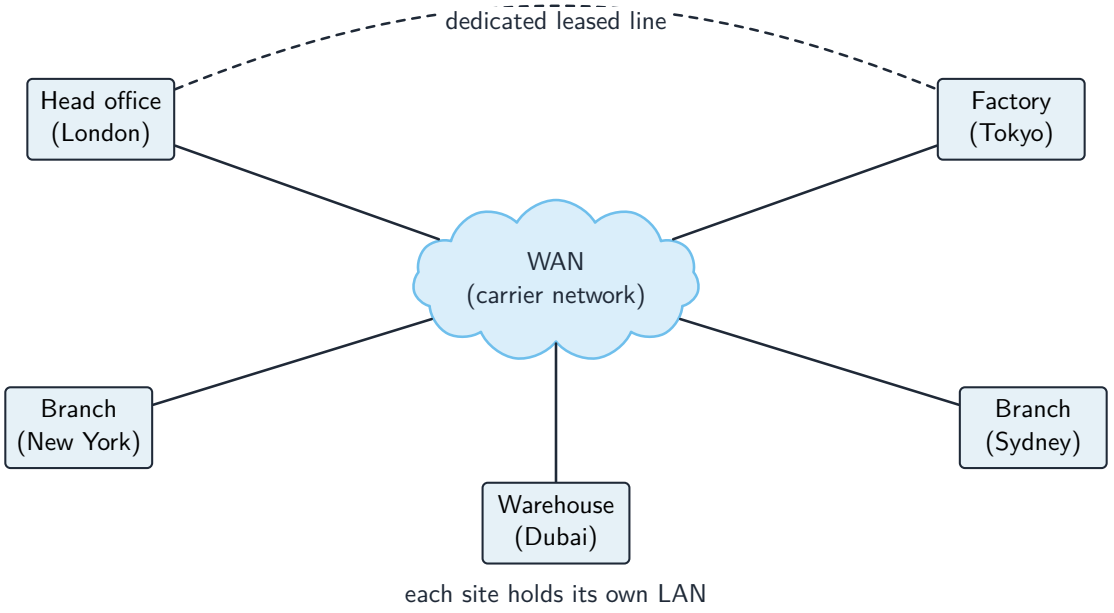
- **sharing resources** (printers, file servers, internet) – cheaper than equipping each computer.
- **sharing data** – many users access the same files.
- **central management** – install software, manage users and back up once on a server.
- **communication** – email, video calls, messaging.
- **remote access** – work from anywhere.

LAN vs WAN

A **local area network** (LAN) covers a **small area** – a home, office or school, usually owned by the organisation, with high data rates and low **latency**.

A **wide area network** (WAN) covers a **large area** – a city, country, or the world (the internet is the largest WAN). It uses telecom-company infrastructure – often the **Public Switched Telephone Network** (PSTN), leased lines or fibre – with lower data rates and higher latency. A WAN connects LANs together.

For "give two characteristics of a LAN": it covers a small geographical area (one site or building); the hardware is owned by the organisation, not leased from a telecom company; it connects through its own switches, cables and access points. For "two ways a WAN is different": it covers a large geographical area; it uses third-party (leased or public) infrastructure; data rates are lower and latency higher; it usually joins several LANs. A school on one site is a LAN; a company with offices in two cities needs a WAN, with a leased line or the internet between the sites. Justify the choice with the area covered and who owns the links.

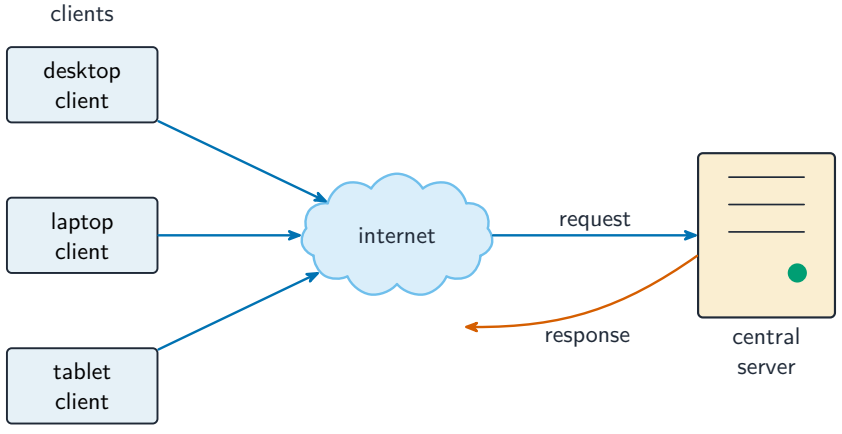


A wide-area network links many systems across a large area

Client-server and peer-to-peer

Client-server

- powerful machines act as **servers** , providing services (files, web pages, email).
- other machines are **clients** that request services.
- central and easy to manage, but the server is a single point of failure unless backed up.

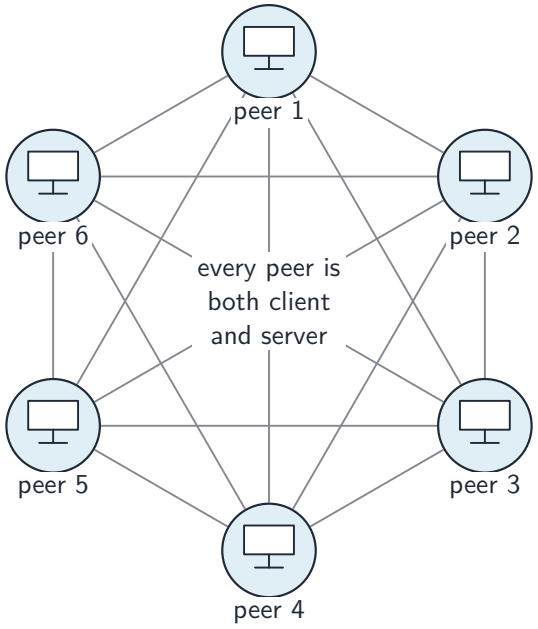


In a client-server network, clients request services from a central server

Peer-to-peer (P2P)

- all machines are equal **peers**; each can be both client and server (**peer-to-peer**).
- resources are spread across the peers – no central server. Robust to one failure, but harder to keep secure and consistent.

Choosing a model. Client-server suits a school or a business: files are stored and backed up centrally, a user logs in with one account from any machine, software and security are managed once, and the server can be a powerful machine. The drawbacks are the cost of the server and of a technician, and that the server is a single point of failure. Peer-to-peer suits a few friends sharing files or a game: no server to buy, easy to set up, and each user keeps control of their own machine. The drawbacks the scheme lists: files are spread across many machines, so they are hard to back up and a file is unavailable when its owner's machine is off; each machine must be secured separately; and a peer that serves the others slows down. An online game played through a web browser with other users is the client-server model: the browser is the client, and the game and its shared virtual world run on the company's server, which keeps every player's view consistent.



In a peer-to-peer network, every node is both client and server

Thin and thick clients

A **thin client** does **little processing locally** and relies on a powerful server (web terminals, remote desktops). A **thick client** has **strong local processing and storage** and runs full applications itself (a normal desktop PC).

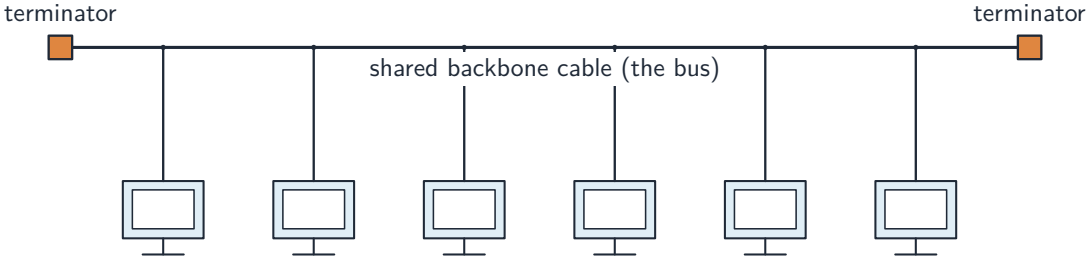
Feature	Thin client	Thick client
Local processing	minimal	substantial
Local storage	minimal	substantial
Reliance on network	high	lower
Server load	high	lower

The roles: in a thin-client model the server does the processing and stores the data, and the client only sends input and shows the output. A cheap terminal is enough, and everything is backed up and updated on the server, but nothing works if the network or the server fails. In a thick-client model the client runs the software and stores files itself, so it can work with no network connection and puts less load on the server, at the cost of more powerful (and more expensive) clients that must each be updated and secured. A school computer room can run thin clients (cheap, centrally managed); a video editor needs a thick client.

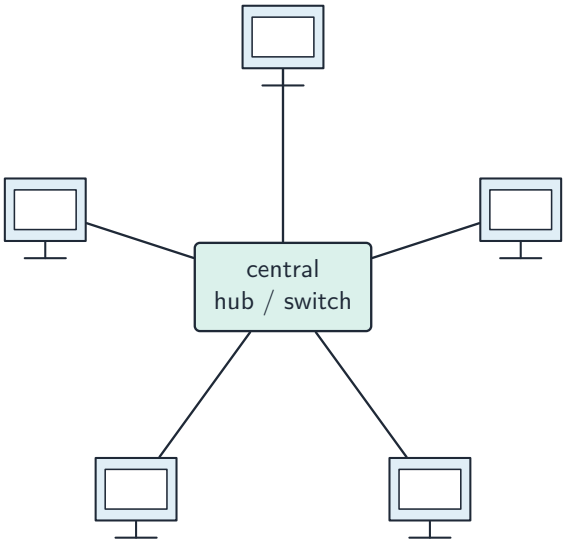
Network topologies

The **topology** is how the nodes and links are arranged.

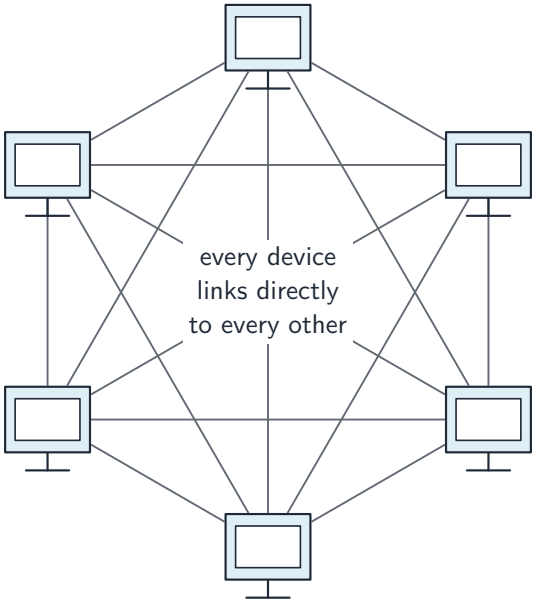
- **bus** – all devices on one shared cable. Cheap; the whole LAN fails if the bus fails; performance drops as more devices share the **bandwidth** .
- **star** – every device connects to a central switch. One device failing does not affect others; the switch failing brings all down. Most common today.
- **mesh** – every device links directly to others, with many paths. Very **fault-tolerant** (traffic reroutes) but needs lots of cabling.
- **hybrid** – a mix (a star in each office, mesh links between offices).



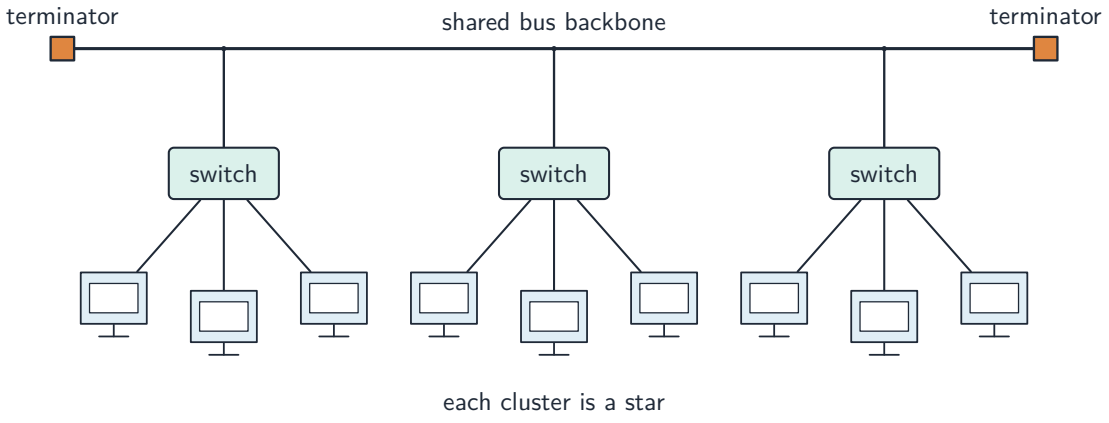
Bus topology: all devices share one cable with a terminator at each end



Star topology: every device connects to a central hub or switch



Mesh topology: every device links directly to the others



Hybrid topology: star clusters joined by a central bus

How packets travel in each topology

Bus: the sending device puts the packet on the shared cable; every device sees it, and only the one whose address matches accepts it. Only one device can transmit at a time, so collisions happen (CSMA/CD, below). **Star**: the sender passes the packet to the central switch, which reads the destination address and forwards it only down the cable to that device; two other devices can talk at the same time. **Mesh**: the packet is passed from node to node along one of several possible routes until it reaches the destination; if a link fails, another route is used.

To justify a topology: a star for a classroom or an office (a failed cable affects one device; a device is easy to add; with a switch there are no collisions); a mesh where reliability matters most (a hospital, the internet's backbone); a bus only where cost matters and few devices share it. "Draw the star topology" means: the switch in the middle, one line from the switch to each computer, and the server (and the router, if there is one) on their own lines to the switch.

Cloud computing

Cloud computing delivers computing services (servers, storage, software) **over the internet**, hosted by a third party. Benefits: **scalability** (pay for what you need), lower cost, access from anywhere, and reliable redundant data centres. Drawbacks: needs internet, your data is held by a third party, and possible vendor lock-in.

For the one-mark definition: cloud computing is **on-demand computing services (storage, processing, software) provided over the internet by a third party**. A **public cloud** is owned by a provider and shared by many customers over the internet; a **private cloud** is dedicated to one organisation, on its own hardware or hosted for it alone. Benefits the scheme accepts: files are accessible from any device with an internet connection; storage scales up and down as needed; the

provider handles the hardware, backups and security updates; there is no local server to buy or maintain. Drawbacks: no access without an internet connection; the data is on a third party's hardware, so security and privacy depend on the provider; an ongoing subscription cost; the provider could fail or be attacked; large files may be slow to transfer. A "why does the company use a public cloud" answer says that they need no hardware of their own, pay only for what they use, and their users can reach it from anywhere.

Wired vs wireless

- **wired** (**Ethernet** over **twisted-pair** or **fibre-optic**): higher speed, lower latency, fewer errors, more secure.
- **wireless** (Wi-Fi, Bluetooth, cellular): no cables, devices can move, but slower, prone to interference and eavesdropping.

For the same generation, wired wins on speed and reliability; wireless wins on convenience.

Transmission media

Medium	Characteristics
copper cable (twisted pair, coaxial)	cheap and easy to install; carries an electrical signal; affected by electromagnetic interference; the signal weakens with distance, so repeaters are needed; lower bandwidth than fibre
fibre-optic cable	light pulses in a glass core; very high bandwidth; long distances without repeaters; immune to interference; hard to tap, so secure; expensive and needs skilled installation
radio waves (including WiFi)	no cable, so devices can move; a range of tens of metres, weakened by walls; a shared frequency, so interference and lower speed; can be intercepted, so needs encryption
microwaves	higher-frequency radio for point-to-point links; needs a line of sight; affected by rain and buildings; high bandwidth
satellites	reach remote areas and the whole globe; a long delay (latency), because the signal travels to orbit and back; affected by weather; expensive

The exam asks for the comparison in both directions. Wired beats wireless on speed, reliability (no interference), security (a cable must be physically tapped) and consistency; wireless beats wired on mobility, the cost of installation, and adding a device without cabling. Allowing both lets students move around with laptops and phones while the fixed desktops keep the faster, more secure connection, and a device with no network port can still connect. Satellite instead of copper reaches places no cable can, but with more delay, weather interference and higher cost.

LAN hardware

- **network interface card** (NIC) – lets a device send and receive on the network; has a unique **MAC address** MAC (a 48-bit hardware address). A wireless device uses a **wireless network interface card** (WNIC).
- **switch** – forwards Ethernet frames only to the port for the destination MAC address.
- **hub** – a simpler device that copies traffic to all ports (now obsolete).
- **wireless access point** (WAP) – lets wireless clients join a wired LAN.
- cabling – twisted-pair for short runs; fibre-optic for longer, faster runs.
- **server** – a computer that provides a service to the other devices: files, printing, web pages, email storage.
- **bridge** – joins two LAN segments into one network, passing traffic between them.
- **repeater** – receives a weakened signal and retransmits it at full strength, to extend a cable's reach.

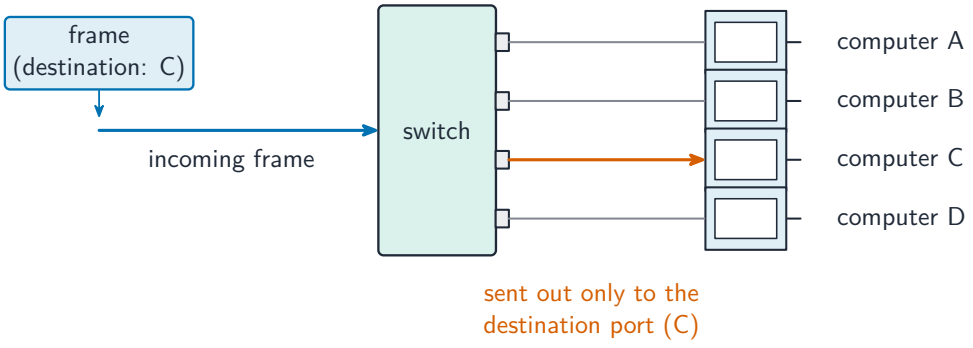
A WNIC's functions, for a four-mark describe: it converts the data into radio signals and back; it carries the device's unique MAC address; it connects the device to a wireless access point and follows the wireless protocol (which channel and frequency to use); and it decodes the incoming signals for the device. Two devices that can physically connect thirty computers with NICs: a switch, or a hub.



A network switch: each device's cable plugs into one of its ports



An RJ-45 plug on a twisted-pair Ethernet cable



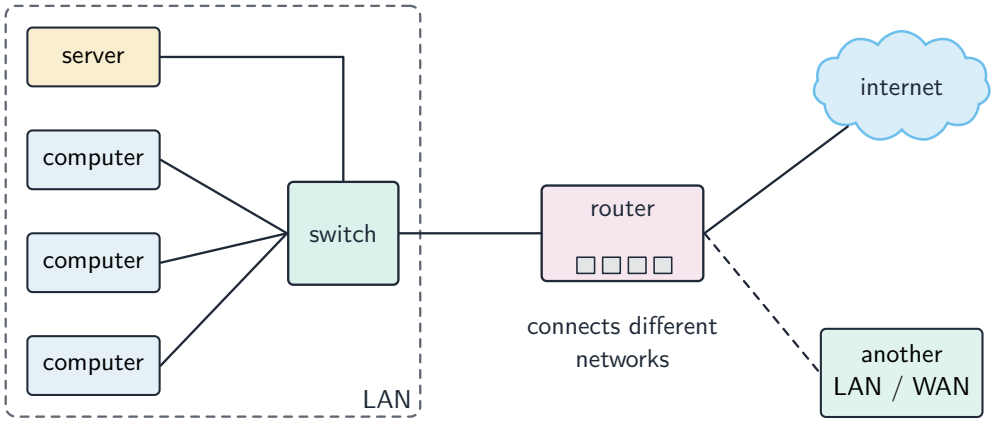
A switch sends each frame only to the port for its destination

Routers

A **router** connects **different networks** and forwards data between them – usually at the boundary of a LAN and the internet. It does:

- **forwarding** – reads each **packet** 's destination **IP address** IP and sends it out the right port, using a **routing table** .
- **network address translation** (NAT) – lets many private LAN addresses share one public IP.
- **DHCP** – hands out private IP addresses to LAN devices.
- **firewall** – blocks unwanted incoming traffic.

In **packet switching** a message is split into packets that are sent independently. Each router reads a packet's destination IP address, looks up the next hop in its routing table and forwards it, so the packets of one message may take different routes and are reassembled in order at the destination. A router does receive packets, forward them between networks and hand out IP addresses; it does **not** find the IP address for a URL (DNS does that) and it does not store web pages. A home router also contains the modem and the wireless access point, so one box connects the LAN to the internet.



A router connects a LAN to the internet or another network

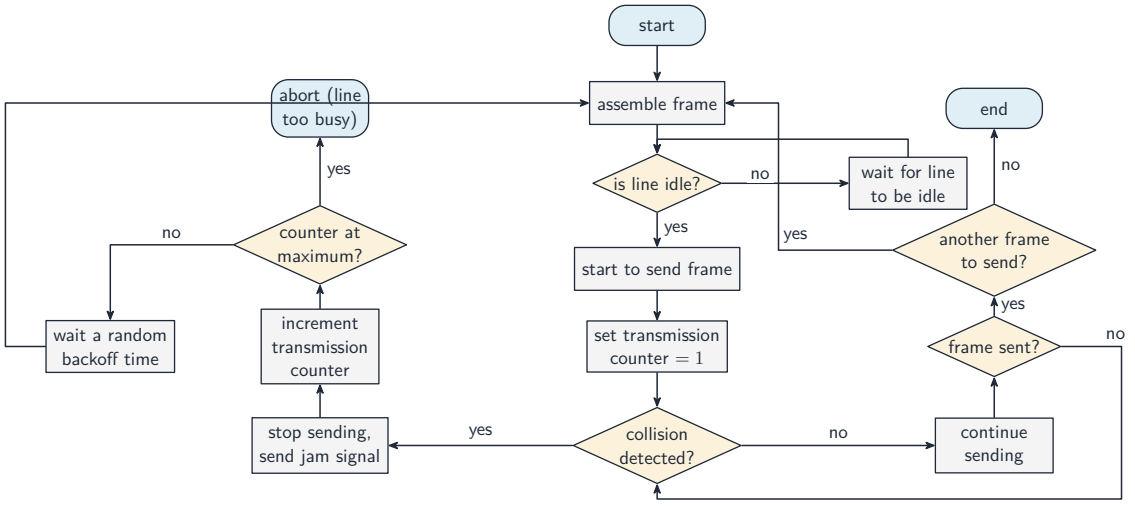
Ethernet and CSMA/CD

Ethernet is the standard (protocol) for wired LANs: devices are joined by twisted-pair or fibre cable, data is sent in frames that carry the source and destination MAC addresses, and a shared medium uses CSMA/CD to deal with collisions. On shared media a **collision** can happen when two devices send at once. The protocol is **CSMA/CD** / (Carrier Sense Multiple Access with Collision Detection):

1. **carrier sense** – listen before sending; wait if the cable is busy.
2. **multiple access** – many devices share the medium.
3. **collision detection** – keep listening while sending; a clash is a collision.
4. on a collision, both stop, send a brief "jam" signal, then wait a **random backoff** time before retrying.

The three tasks, in the scheme's words: the device listens (senses the carrier) before transmitting; it keeps checking for a collision while it transmits; on a collision it stops, sends a jam signal, waits a random time and retransmits.

Modern switched Ethernet uses **full-duplex** point-to-point links, so collisions no longer happen.



The CSMA/CD process for handling collisions on shared media

Bit streaming

Bit streaming sends multimedia as a continuous stream that the receiver plays as it arrives, instead of downloading the whole file first.

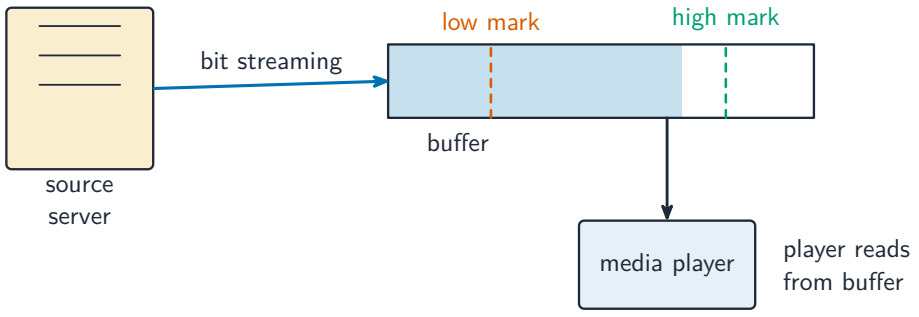
- **real-time** (live): captured and streamed as it happens (live sport, video calls). You cannot rewind; low latency is vital.
- **on-demand**: pre-recorded on a server (YouTube, Netflix). You can pause and rewind; the server can **buffer** ahead.

Real-time streaming works as a short pipeline:

1. **capture and sample** the source (a camera or microphone).
2. **encode** it, using **compression** to shrink the data.
3. **send** it across the network as packets.
4. the receiver buffers a little, then plays it live – **dropping any packet that arrives late**, because a live stream cannot wait for it.

Lossy compression is used here: moving pictures hide small losses, and the stream must be small enough to fit the bandwidth.

Why a video is compressed before real-time streaming: the uncompressed stream would need more bandwidth than the connection has, so frames would arrive late and the playback would stall. Compression cuts the number of bits, so the **bit rate** stays below the broadband speed, the delay stays small, and less storage and cost are needed at both ends. The bit rate must be lower than the connection's speed: a higher bit rate gives better quality but needs a faster connection, and if the data arrives more slowly than it is played, the buffer empties and the video freezes. On-demand streaming can buffer more of the file ahead, so it copes with a slower connection; real-time streaming cannot.

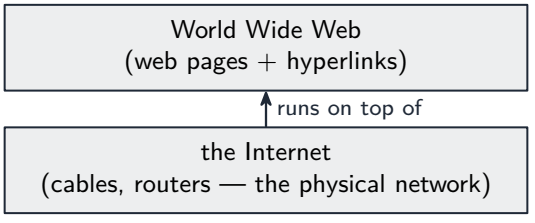


Data streams from the server into a buffer before the media player reads it

The internet and the World Wide Web

The **internet** is a global **network of networks** using a common **protocol** suite (TCP/IP). The **World Wide Web** (WWW) is a **service** that runs over it: hyperlinked documents identified by URLs, viewed in browsers via HTTP/HTTPS. Email and file transfer are other internet services that are not part of the WWW.

Webmail uses **both**: the WWW, because the mailbox is a web page reached through a URL in a browser over HTTP; and the internet, because the email itself travels across the network of networks (email is a separate internet service from the web).

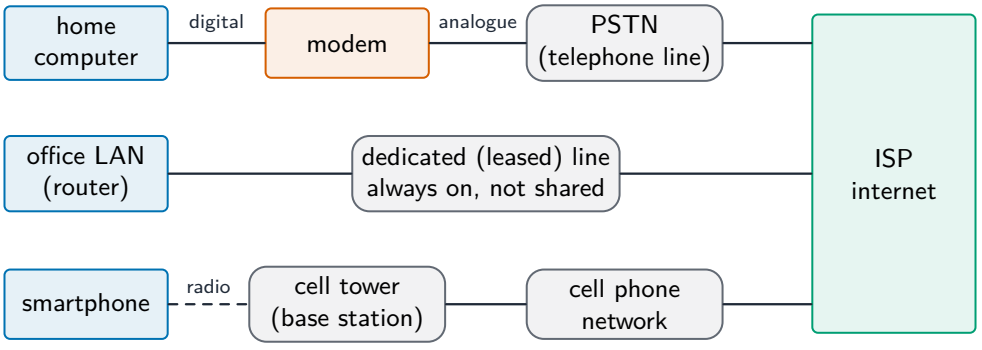


The Web is one service running on top of the Internet

Hardware that supports the internet

- **modem** – converts the computer's digital signal into an analogue signal for a telephone line, and back again at the other end (modulation and demodulation).
- **PSTN** – the public telephone network of exchanges and lines; a dial-up or DSL connection carries internet data over it.
- **dedicated line** – a leased line between an organisation and its ISP: always on, with a fixed bandwidth that is not shared, so faster and more reliable, but expensive.

- **cell phone network** – the phone sends data by radio to the nearest cell tower (base station); the towers are linked to the phone company's network, which routes the data to the internet; as the phone moves, it is handed over from one cell to the next.



Three ways to reach the internet: a modem and the PSTN, a dedicated line, and the cell phone network

IP addresses

An IP address uniquely identifies a device.

- **IPv4** – 32-bit, four denary numbers 0–255 (`192.168.1.10`); about 4.3×10^9 addresses (now exhausted).
- **IPv6** – 128-bit, eight groups of four hex digits; about 3.4×10^{38} addresses.

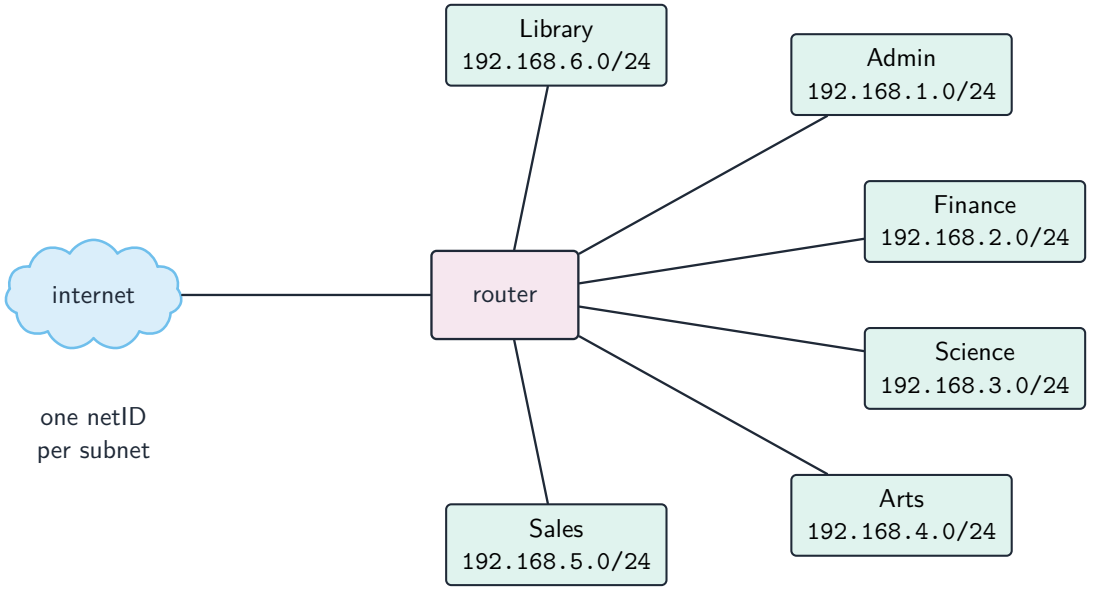
IPv4 is written as **four** groups of denary numbers separated by **dots**; each group is an 8-bit number, so it runs from **0 to 255**. IPv6 is written as **eight** groups of **four hexadecimal digits** separated by **colons**, `2001:0db8:0000:0000:0000:ff00:0042:8329`, and a run of zero groups can be shortened to `::`. So `192.168.3.2` is not IPv6: it has four groups, not eight, separated by dots rather than colons, and its groups are denary, not hexadecimal. `256.0.0.A` is not a valid address of either kind: an IPv4 group cannot exceed 255 and cannot be a letter, and IPv6 would need colons and eight groups.

■ Subnetting

A network can be split into **subnets** . The IP address splits into a network part and a host part, given by a **subnet mask** (e.g. `255.255.255.0` = first 24 bits are network). Subnetting improves management, cuts broadcast traffic, and improves security.

The two parts of an address in a subnetwork: the **network ID** (the first bits, the same for every device in that subnet, given by the ones in the mask) and the **host ID** (the remaining bits, unique to each device). Benefits of subnetting, for "describe two benefits": less traffic on each part, because broadcasts stay inside their subnet; better

security, because one department's traffic is kept from the others; easier management and fault-finding; more efficient use of the addresses. Two devices with the mask `255.255.255.0` are in different subnets when their first three groups differ.



Splitting a network into subnets, one netID per department

Public vs private addresses

- **private** addresses are used within a LAN and are not routable on the internet (e.g. `192.168.0.0/16`).
- a **public IP address** is globally unique and routable, assigned by an **ISP**

.

Devices behind NAT with private addresses are not directly reachable from the internet, giving some protection.

The descriptions the tables want: a **public** address is visible on the internet and unique across it, allocated by the ISP; a **private** address is visible only inside the LAN, is reused by many LANs, and needs NAT to reach the internet. A **static** address never changes (set by hand or reserved, as a server needs); a **dynamic** address is allocated by DHCP each time the device connects and may change.

Static vs dynamic

- a **static IP address** is fixed; used for servers that must be found at a known address.
- a **dynamic IP address** is assigned by DHCP and may change; easier for client devices and uses a limited address pool efficiently.

Worked example. A host has IP address `192.168.10.130` with subnet mask `255.255.255.192`. Which network is it on, and is `192.168.10.200` on the same one? The mask's last octet, `192`, is `11000000` in binary, so the first **26** bits are the network part and the last 6 bits address the host. That makes the subnets step in blocks of $256 - 192 = 64$: `.0`, `.64`, `.128`, `.192`. The address `130` falls in the block starting at `.128`, so the host is on network **`192.168.10.128/26`**, whose usable hosts run `.129` to `.190` (`.191` is the broadcast address). `200` falls in the **next** block (`.192`), so it is on a **different** subnet and traffic between the two must pass through a **router**. Get the block size from the mask first (256 minus the mask octet) - guessing from the first three octets is what makes these go wrong.

URL and DNS

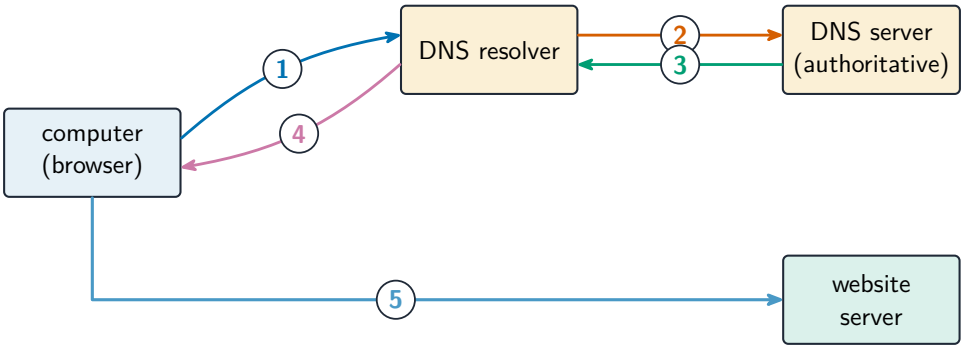
A **URL** (Uniform Resource Locator) locates a resource on the WWW:

```
https://www.example.com/about/contact.html
protocol     domain name        path
```

- **protocol**: `http`, `https`, etc.
- **domain name** : a readable server address.
- **path**: the resource on that server.

The **Domain Name System** (DNS, also called the Domain Name Service) is a distributed set of servers that turns **domain names into IP addresses**. When you type a URL, the browser asks a DNS resolver for the IP, which queries DNS servers (root → top-level → authoritative) until it finds it; the browser then connects to that IP and requests the path. DNS saves humans from memorising IP addresses and lets a site change server without changing its name.

For "explain how the browser uses the URL": the browser splits the URL into the protocol, the domain name and the path; it sends the domain name to a DNS server, which returns the matching IP address (a cache on the computer or at the ISP may answer first); it opens a connection to that IP address using the protocol (HTTPS on port 443); it sends a request for the path; and the web server returns the page, which the browser renders. If the DNS lookup fails, the browser reports that the server cannot be found.



1 computer asks the resolver for the IP **2** resolver queries a DNS server
3 server returns the IP address **4** resolver passes the IP back to the computer
5 browser connects to that IP and requests the page

How DNS finds a website's IP address before the browser connects

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly.

Term	Definition
LAN	a network covering a small geographical area, usually one site, whose hardware is owned by the organisation
WAN	a network covering a large geographical area, joining LANs through third-party (leased or public) links
client-server	a model in which client computers request services from a central, more powerful server that provides them
peer-to-peer	a model in which every computer is equal and can act as both client and server, with no central server
thin client	a client that does little processing or storage itself and depends on the server for both
thick client	a client that does its own processing and storage and can work without the server
mesh topology	a topology in which each device is connected directly to many others, giving more than one route between two devices
cloud computing	on-demand computing services (storage, processing, software) provided over the internet by a third party
Ethernet	the standard protocol for wired LANs, sending data in frames and using CSMA/CD on a shared medium

Term	Definition
switch	a device that forwards each frame only to the port of its destination MAC address, within a LAN
router	a device that connects networks and forwards packets between them by their destination IP address
bit streaming	sending a continuous stream of bits so that the receiver plays the media as it arrives, without downloading the whole file first
internet	the global network of networks that uses the TCP/IP protocols
World Wide Web	the collection of hyperlinked web pages, identified by URLs and accessed over the internet through a browser
URL	the address that locates a resource on the web: protocol, domain name and path
DNS	the service that translates a domain name into the IP address of the server that holds the resource

Exam tips

- Distinguish **LAN vs WAN** and **client-server vs peer-to-peer** by who stores and controls the resources.
- Match each **topology** (bus, star, mesh) to its advantages and drawbacks (cost, reliability, collisions).
- Know the job of each device: a **switch** directs within a LAN by MAC address, a **router** routes between networks by IP.
- Explain **bit streaming** and why buffering is needed (data arrives at a different rate from playback).
- Distinguish **IPv4 vs IPv6** and public vs private addresses; DNS turns a URL into an IP address.

Common mistakes

- Saying a switch works by IP address. A switch forwards by MAC address inside the LAN; the router forwards by IP address between networks.
- Treating the internet and the World Wide Web as the same thing. The web is one service that runs over the internet; email and file transfer are others.
- Giving "faster" as the whole comparison of wired and wireless. Say faster **and** more reliable and more secure, and give the wireless side (mobility, no cabling) when the question asks for a comparison.

- Writing that a router finds the IP address for a URL. DNS does that; the router forwards packets to it.
- Describing IPv6 with dots and denary groups. Eight groups of four hexadecimal digits, separated by colons.
- Drawing a star topology as a ring or a chain. Every device has its own line to the switch in the middle.

Hardware

A-Level Computer Science

Computers and their components

A general-purpose computer has four building blocks:

- **input devices** – get data in (keyboard, mouse, microphone, scanner, sensors).
- **output devices** – give results out (monitor, speakers, printer, actuators).
- **primary memory** – fast memory the **processor** (CPU) reaches directly (RAM and ROM). Holds the running program and its data.
- **secondary storage** – slower, larger, keeps programs and data when not in use (hard disk, SSD, optical disc, USB stick).

The syllabus asks why each is **needed**. Input devices are needed because the computer can only work on data and instructions that have been entered. Output devices are needed to present the results in a form people can use. Primary memory is needed because the processor can only execute instructions and use data that are held in memory it can address directly, and it must reach them fast. Secondary storage is needed because primary memory is volatile and small: programs and data must survive the power being switched off, in a larger and cheaper store, and **removable** storage lets data be moved between computers or kept as a backup.



A keyboard: a common input device for typing text and commands



A mouse: a pointing input device



A flatbed scanner: an input device that turns a paper page into a digital image



A monitor: a common output device that displays the screen image

Embedded systems

An **embedded system** is a computer built **into another device** to do one fixed job (washing machine, microwave, car engine unit, thermostat).

- **benefits**: optimised for one task, so it is small, uses little power and is cheap to make in volume; reliable, because it runs one fixed program with few chances to go wrong; starts quickly and needs no user set-up; easy to use through a simple interface.
- **drawbacks**: limited to its one task, so it cannot be upgraded to do more; hard to update (its **firmware** may need special tools or cannot be changed at all); difficult to troubleshoot, and usually the whole device must be replaced when it fails; if it is connected to a network it can be a security weakness, because its software is rarely patched.

A "describe the drawbacks" question wants each drawback as a full point: what the limitation is and what it means for the user, for example "the firmware cannot be updated, so a security fault found later cannot be fixed".

Principal hardware devices

■ Laser printer

A **laser printer** scans the page image onto a charged photosensitive **drum** . **Toner** sticks to the charged areas, transfers to the paper, and is melted on by a fuser. Fast, sharp, high-volume.



A laser printer: fast, sharp printing using a charged drum and toner

How it works, in the steps the mark scheme lists:

1. The data for the page is sent to the printer's buffer.
2. The **drum** is given a uniform electrostatic charge.
3. A laser, reflected off a rotating mirror, scans the page image onto the drum, removing the charge where it strikes, so the charge left on the drum matches the image.
4. **Toner**, a charged powder, is attracted to the charged parts of the drum only.
5. The paper is given the opposite charge and rolled against the drum, so the toner transfers onto it.
6. The **fuser** , a pair of heated rollers, melts the toner into the paper. The drum is then discharged and cleaned for the next page.

■ 3D printer

A **3D printer** 3D builds an object **layer by layer**: FDM melts plastic filament through a nozzle; stereolithography cures liquid resin with a UV laser. Used for prototypes and custom medical parts.



An FDM 3D printer builds an object layer by layer by melting plastic filament

How it works:

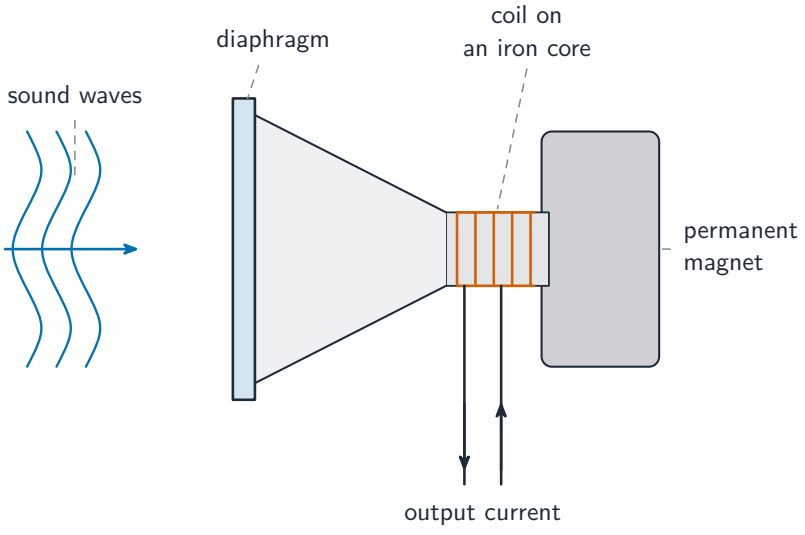
1. A model of the object is designed in **CAD** software (or scanned).
2. Slicing software divides the model into thin horizontal layers and produces the instructions for each one.
3. The printer builds the object one layer at a time: an FDM printer melts plastic **filament** and lays it down through a moving nozzle; a resin printer cures liquid resin with a laser or UV light; a powder printer fuses powder with a laser.
4. Each layer bonds to the layer below, and the platform (or nozzle) moves by one layer's thickness.
5. When the last layer is done, any support material is removed. Uses include prototypes, custom medical parts such as prosthetics, and spare parts printed on demand.

■ Microphone and speakers

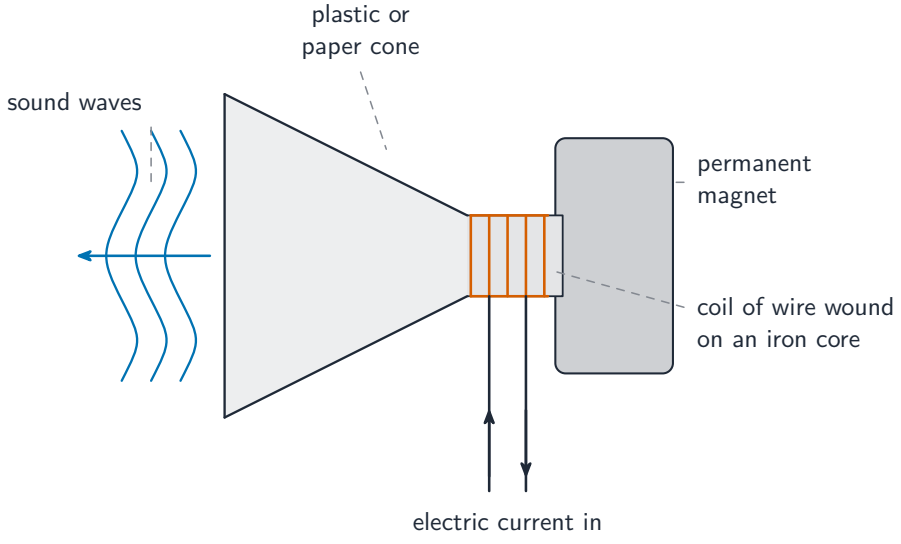
A **microphone** turns **sound into an electrical signal** (a diaphragm vibrates, changing **capacitor** charge or coil position); the signal is digitised by an **analogue-to-digital converter** (ADC). A speaker does the reverse – a varying signal drives a coil in a magnetic field, moving a cone to make sound.



A microphone turns sound into an electrical signal



Inside a microphone: sound vibrates the diaphragm and coil to produce a current



Inside a loudspeaker: a varying current in the coil moves the cone to make sound

How a microphone works: sound waves make a **diaphragm** vibrate; in a dynamic microphone a coil attached to the diaphragm moves in a magnetic field, so a varying current is induced in it, and in a condenser microphone the diaphragm is one plate of a capacitor whose capacitance changes as it moves; the varying analogue signal is then sampled by an ADC and stored as digital data. A speaker runs the chain backwards: a **digital-to-analogue converter** (DAC) produces a varying current, the current in the coil creates a changing magnetic field that pushes against the permanent magnet, the coil and cone move in and out, and the cone's movement makes pressure

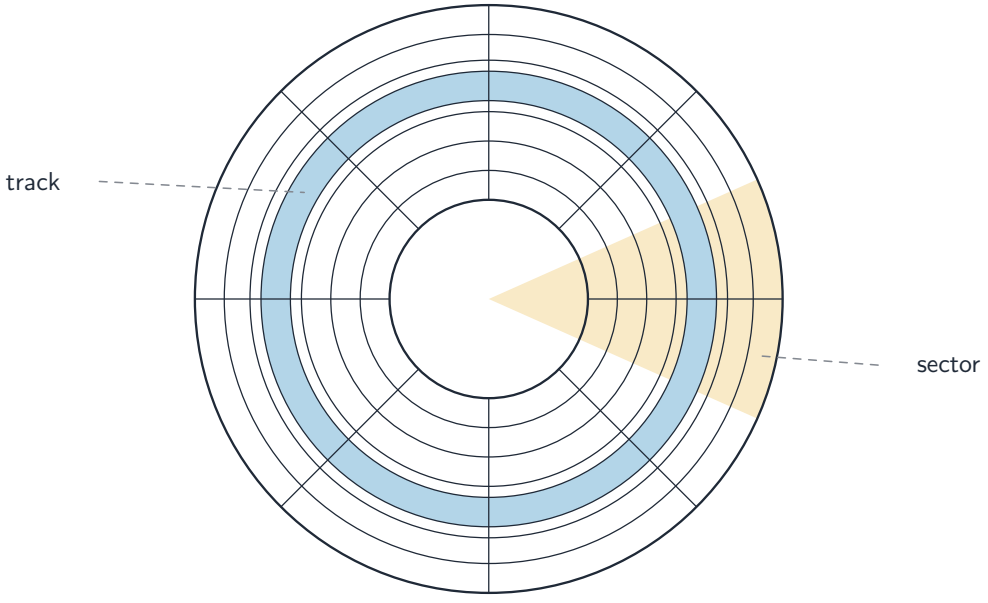
waves in the air.

Magnetic hard disk (HDD)

A **hard disk** stores data on spinning platters coated with magnetic material. Each platter has **tracks** divided into **sectors**. A **read/write head** floats just above and magnetises tiny regions (write) or senses them (read). Cheap per gigabyte, but slower than SSDs and has moving parts.



An opened hard disk: the actuator arm carries the read/write head over a platter



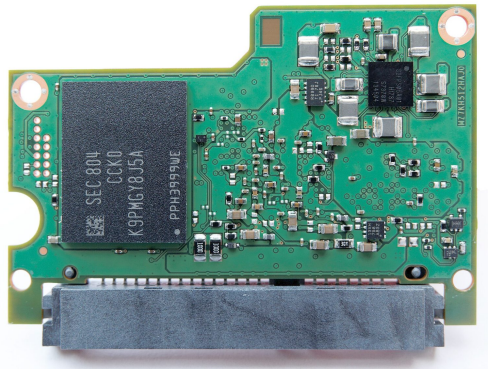
Tracks and sectors on a hard disk platter

How it works: the platters spin at high speed (thousands of revolutions per minute); each surface is divided into concentric tracks and each track into sectors; read/write heads on **actuator arms** move across the platters to the right track; to write, the head magnetises a tiny region with one of two polarities, representing 0 or 1; to

read, it detects the polarity as the region passes beneath it. The delays, waiting for the arm to reach the track and for the sector to spin round, are why a hard disk is slower than an SSD.

■ Solid-state (flash) memory

A **solid-state drive** stores data as charge in **transistors**, with no moving parts. Faster random access than HDDs, tougher, lower power, but dearer per gigabyte; each cell wears out after many writes.



Inside an SSD: data is stored in flash memory chips, with no moving parts (compare the hard disk above)

How it works: each cell is a **floating-gate transistor**; a charge trapped on the floating gate represents a bit and stays there when the power is off; a controller chip maps each address to a cell and spreads writes across the cells, because a cell survives only a limited number of writes.

	Magnetic hard disk	**Solid-state drive**
Moving parts	platters and heads	none
Speed	slower: seek and rotation delays	much faster random access
Cost per gigabyte	lower	higher
Robustness	damaged by knocks; noisy; more power	shock-resistant; silent; less power
Lifetime	many rewrites; wears mechanically	limited write cycles per cell

A "why a server uses hard disks rather than SSDs" question wants the left column: cheaper per gigabyte for very large capacities, a long life under constant rewriting, and easier data recovery.

Optical disc

A laser detects reflections from tiny pits on an **optical disc** (CD, DVD, Blu-ray). The drive is an **optical disc reader/writer**: writing uses a stronger laser to change the surface's reflectivity.

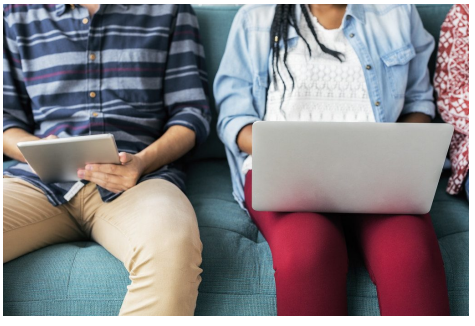


An optical disc drive: a laser reads tiny pits on a CD, DVD or Blu-ray disc

How it works: the disc carries one long spiral track of **pits** and **lands** (the flat areas between them); the disc spins and a laser is focused on the track; light reflected from a land differs from light reflected at the edge of a pit, and a light sensor reads each change as a 1 and no change as a 0. Writing uses a stronger laser to change the reflectivity of a dye or alloy layer. A Blu-ray uses a blue laser with a shorter wavelength, so its pits are smaller and closer together, which is why it holds more data than a DVD.

Touchscreen

A **touchscreen** senses contact. **Resistive** : two conductive layers pressed together; works with anything but is less accurate. **Capacitive** : a finger disturbs a charge field; accurate, multi-touch, used in phones.



A touchscreen senses where a finger touches the glass

How it works: a resistive screen has two thin conductive layers separated by spacers; pressing pushes the top layer onto the bottom one, closing a circuit at that point, and the controller reads the voltage to find the coordinates. A capacitive screen has a

glass layer coated with a transparent conductor that holds a charge; a finger touching it draws a tiny current, the current is measured at each corner, and the controller works out the touch position from the differences. Capacitive screens respond to a light touch and to several fingers at once, but not to a gloved finger or an ordinary stylus.

■ Virtual reality headset

A **virtual reality** (VR) headset has two small displays (one per eye) and motion sensors (**accelerometer** , **gyroscope**) that track head movement so the scene shifts as you look around.



A virtual reality headset: two small displays and motion sensors track the head

How it works: each eye sees its own display through a lens, and the two images differ slightly, so the brain sees depth; sensors (accelerometer, gyroscope, sometimes cameras) report where the head is and which way it points; the computer re-renders the scene from that viewpoint many times a second, so turning the head turns the view; headphones give sound that matches the direction. Used for games, for training such as flight or surgery simulators, and for viewing designs before they are built.

Buffers

A **buffer** is memory that **holds data temporarily** while it moves between devices of different speeds. Example: the CPU writes a document to a printer buffer quickly, then is free to do other work while the printer prints from the buffer at its own pace. Buffers stop the fast device waiting for the slow one (also used in streaming, the keyboard, and disk access).

"State why a 3D printer needs a buffer": the computer sends the print data much faster than the printer can build the layers, so the data is held in the buffer until the printer is ready for it, and the processor is freed to do other work. When the buffer runs low the printer sends an **interrupt** to ask for more (topic 4). A video stream works the same way: the buffer fills ahead of playback so a short drop in the network speed does not stop the picture.

RAM and ROM

- **RAM** (Random Access Memory) – **volatile** (loses data without power). Holds the OS, running programs and their data; read and written constantly.
- **ROM** (Read-Only Memory) – **non-volatile** (keeps data without power). Usually written once; holds firmware needed at start-up (the BIOS / boot loader).

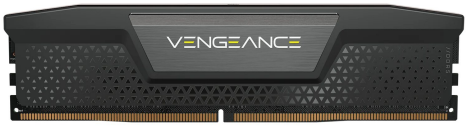


RAM is volatile and read/write; ROM is non-volatile and read-only

ROM starts the system; RAM then holds the active work.

	RAM	**ROM**
Volatile?	yes: contents lost when the power is off	no: contents kept without power
Read/write?	read and written constantly	read only in normal use
Holds	the operating system, running programs and their data	the firmware and bootstrap program that start the computer
Size	large, and can usually be increased	small and fixed
Typical use	the main memory of a computer or phone	the start-up code of a PC; the whole program of an embedded system such as a washing machine

More RAM lets a computer hold more programs and data at once, so it swaps less between memory and disk and runs faster; that is the answer to "explain why the computer with more RAM performs better".



A RAM module (DIMM) plugs into the motherboard as the computer's fast main memory

The same memory split matters in a wearable device: its fixed program must remain available after power off, while its live readings change during use.



A wrist-worn heart-rate monitor

SRAM vs DRAM

- **SRAM** RAM (Static RAM) stores each bit in a **flip-flop** of several transistors. Fast, but expensive and not dense. Used for CPU **cache** .
- **DRAM** RAM (Dynamic RAM) stores each bit as charge on a tiny capacitor. Cheaper and denser but slower, and must be **refreshed** (rewritten) thousands of times a second. Used for main memory.

Use SRAM for small fast memory (cache); DRAM for large main memory.

	SRAM	DRAM
Each bit stored in	a flip-flop of several transistors	one capacitor and one transistor
Needs refreshing?	no	yes, thousands of times a second
Speed	faster	slower
Density and cost	fewer bits per chip, more expensive	more bits per chip, cheaper
Power	uses less power when idle	uses more, because of the refresh
Used for	processor cache	main memory, including in embedded systems

"Explain why the embedded system uses DRAM": it needs a large amount of memory at low cost in a small space, and its speed requirement is modest, so the cheaper, denser DRAM is the right choice; SRAM is kept for the small cache where speed matters most.

PROM, EPROM and EEPROM

ROM variants you can program after manufacture:

- **PROM** (Programmable ROM) – written **once** (fuses burned by a programmer); cannot be changed.
- **EPROM** (Erasable Programmable ROM) – erased by **strong UV light** through a window, then rewritten (whole chip at once).
- **EEPROM** (Electrically Erasable Programmable ROM) – erased and rewritten **electrically**, a byte at a time, in circuit. Flash memory is a derivative optimised for block erase.



An EPROM chip with a window for ultraviolet erasure

	PROM	EPROM	EEPROM
Written	once, by the user with a programmer	many times	many times
Erased by	cannot be erased	ultraviolet light through a quartz window	an electrical signal
Erases	nothing	the whole chip at once	a byte or block at a time
Must be removed from the circuit to reprogram?	not applicable	yes	no

"Give two differences between EPROM and EEPROM" wants two rows of this table, each stated for both types.

Monitoring and control systems

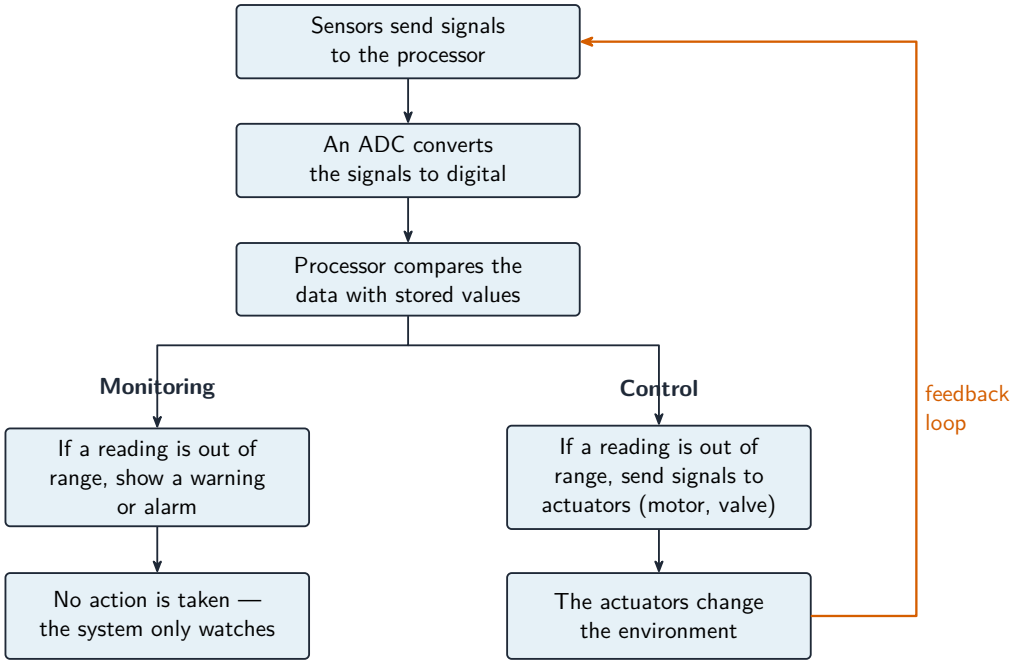
Both read **sensors**; the difference is what they do next.

- **monitoring** – collects and reports data but **takes no action** (a weather station logging readings).
- **control system** – uses sensor data to **decide and act** through actuators, usually in a feedback loop (a thermostat turning a boiler on/off).

The three-mark "describe the differences" answer: a monitoring system only measures, records or displays the readings, and at most raises a warning; a control system compares each reading with a **preset value** and, if it is outside the range, sends signals to actuators that change the physical process; the change is then measured again, so a control system contains feedback and a monitoring system does not. Whether a given system is one or the other is decided by that test: a bridge system that measures a vehicle's height and switches on a warning sign is monitoring, because nothing it does changes the vehicle; a system that lowers a barrier is control.

> **Worked example.** Describe how an automated system opens a door when a person is within 2 metres and closes it when nobody is.

An infra-red or ultrasonic sensor measures the distance to anything in front of the door; the analogue reading is converted to digital by an ADC and sent to the processor; the processor compares the distance with the preset 2 metres; if it is less, the processor sends a signal to the actuator (a motor) to open the door; the sensor keeps measuring, and when no reading below 2 metres is received the processor signals the motor to close the door. The repeated measuring after each action is the feedback that stops the door opening and closing at the wrong times.



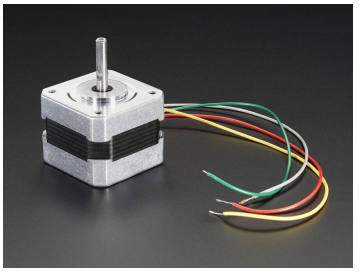
Monitoring reports data; a control system acts through a feedback loop

Sensors and actuators

A **sensor** turns a physical quantity into a signal: temperature (a **thermistor** or thermocouple), pressure (strain gauge), infra-red, sound. Analogue signals need an ADC first. An **actuator** does the reverse – turns a signal into an action (a motor, valve, heater, buzzer).



A thermistor: a temperature sensor whose resistance changes with heat



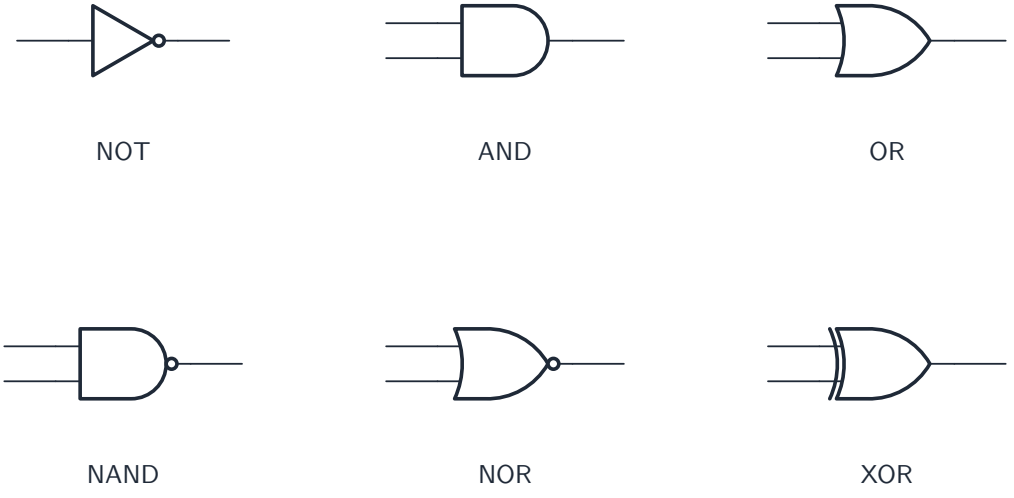
A small electric motor: an actuator that turns a signal into movement

Feedback

In a control system the actuator changes the environment, which the sensors then re-measure – a **feedback** loop. Without feedback the system cannot correct itself or know when to stop (a thermostat with no temperature feedback would heat forever).

Logic gates

A **logic gate** is a small circuit that does one **Boolean** operation. Inputs and outputs are 0 (false, low) or 1 (true, high). Know the symbol, function and **truth table** for each gate.



The symbols for the six logic gates

NOT (inverter)

A	NOT A
0	1
1	0

- **AND – output 1 only if all inputs are 1**

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

- **OR – output 1 if at least one input is 1**

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

- **NAND (NOT AND) – output 0 only when all inputs are 1**

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

- **NOR (NOT OR) – output 1 only when all inputs are 0**

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

- **XOR (Exclusive OR, also called EOR) – output 1 if the inputs are different**

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

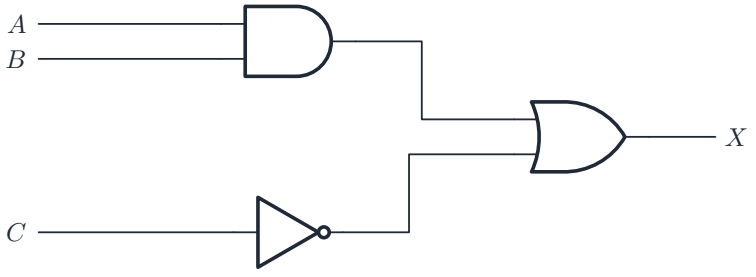
Logic circuits

A **logic circuit** is a network of gates that carries out a Boolean expression. You should be able to move between a problem statement, a logic expression, a truth table, and a circuit diagram.

The paper writes expressions in words, `X = (A AND NOT B) OR (B AND C)`, and accepts the algebraic form $X = A\overline{B} + BC$ where a dot (or nothing) is AND, a plus is OR, and a bar is NOT. Use whichever the question uses.

From expression to circuit

Draw one gate per operator and wire them up. For $X = (A \text{ AND } B) \text{ OR } (\text{NOT } C)$: a NOT gate on C, an AND gate on A and B, then an OR gate on the two results.

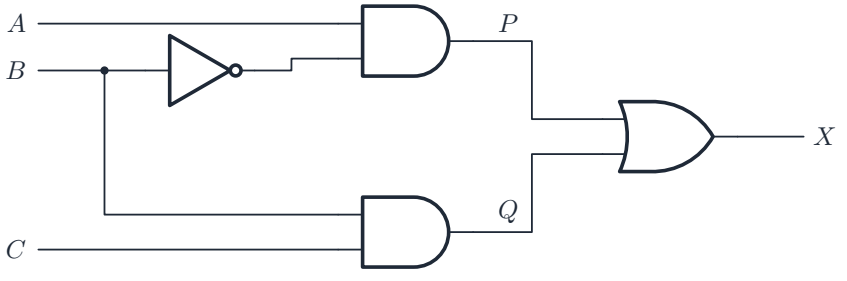


Gates wired together to carry out a Boolean expression

From circuit to expression

Work forwards from the inputs, labelling each gate's output, until you reach the final output.

> **Worked example.** Write the expression for the circuit below, then complete its truth table.



Label every intermediate output: here P is A AND NOT B and Q is B AND C, so X is P OR Q

Label the gate outputs: $P = A$ AND NOT B, $Q = B$ AND C, so $X = P$ OR $Q = (A \text{ AND NOT } B) \text{ OR } (B \text{ AND } C)$. Then give the truth table a column for each intermediate output, so every row can be checked one gate at a time:

A	B	C	NOT B	P	Q	X
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	0	1	1	1	0	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Drawing a circuit from an expression is the same walk in reverse: start from the innermost brackets, draw one gate per operator, draw a NOT gate on the wire of any input that appears with NOT, keep the inputs on the left and the single output on the right, and label the output with its letter. Every line must end at a gate input or the output; a line that goes nowhere loses the mark.

■ From circuit to truth table

For n inputs there are 2^n rows. List every input combination; for each, work out the internal gates then the output.

■ From truth table to expression (sum of products)

For each row that outputs 1, write an AND of the inputs (with NOT on any input that is 0 in that row); OR these together. Example: a table that is 1 only on $(A = 0, B = 1)$ and $(A = 1, B = 0)$ gives $\overline{A}B + A\overline{B}$, which is A XOR B.

■ From a problem statement

Turn the English into a Boolean expression first: “A and B” → A AND B; “A or B or both” → A OR B; “exactly one of A and B” → A XOR B; “neither A nor B” → A NOR B; “not both” → A NAND B.

> **Worked example.** A machine’s alarm X sounds when the guard is open $(A = 1)$ **and** either the motor is running $(B = 1)$ or the temperature is high $(C = 1)$. Write the Boolean expression, and give the rows where $X = 1$. Turn the English into logic one clause at a time: ”either B or C” is $B + C$, and ”A **and** that” is $X = A \cdot (B + C)$. For the rows, $X = 1$ needs $A = 1$ **and** at least one of B, C equal to 1 - so $(A, B, C) = (1, 0, 1)$, $(1, 1, 0)$ and $(1, 1, 1)$, three rows out of eight. Notice $A = 0$ can never sound the alarm, whatever B and C do. **Bracket the OR** before ANDing it: $X = A \cdot B + C$ is a different circuit altogether, one that would sound the alarm on a high temperature even with the guard **closed**.

Flash cells and light sensors

In a floating-gate flash cell, an insulated floating gate traps charge without continuous power. A voltage on the **control gate** is used to program or read the cell; trapped charge changes the voltage threshold at which the transistor conducts, so the read circuit distinguishes stored states. Erasing removes the trapped charge. **NAND flash** connects cells in strings and is suited to dense block storage; **NOR flash** allows direct access to individual locations, useful for firmware. Neither requires moving parts.

A **light sensor** measures light intensity, for example to control a street lamp when daylight falls. Its analogue signal can pass through an ADC before the processor compares it with a threshold. State the measured physical quantity, the conversion if needed, and the resulting control action.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
embedded system	a computer system with a dedicated function built into a larger device
buffer	an area of memory that temporarily stores data while it is transferred between devices working at different speeds
RAM	volatile memory that can be read from and written to, holding the programs and data in use
ROM	non-volatile memory whose contents cannot be changed in normal use, holding the start-up instructions
SRAM	static RAM that stores each bit in a flip-flop and needs no refreshing
DRAM	dynamic RAM that stores each bit as a charge on a capacitor and must be refreshed continually
monitoring system	a system that uses sensors to measure and report on a physical process without changing it
control system	a system that uses sensor readings to decide on and carry out actions, through actuators, that change a physical process
sensor	a device that measures a physical quantity and converts it into a signal for the computer
actuator	a device that converts a signal from the computer into a physical action

Term	Definition
feedback	the output of a control system being measured and fed back as input so that the system can correct itself
logic gate	an electronic circuit that performs a Boolean operation on one or more binary inputs to give one binary output
truth table	a table listing every combination of inputs to a logic circuit with the output for each

Exam tips

- Distinguish **RAM (volatile, read/write)** from **ROM (non-volatile, holds the bootstrap)**; SRAM (cache, faster) from DRAM (main memory, needs refreshing).
- For a **logic circuit**, build the Boolean expression gate by gate, then a truth table covering **every** input combination.
- Learn the symbol, expression and truth table for each gate (AND, OR, NOT, NAND, NOR, XOR).
- Explain a **buffer** (a temporary store bridging two different speeds) and the role of an interrupt.

Common mistakes

- Naming the device instead of describing its operation. "It uses a laser" earns nothing; the steps (charge the drum, laser removes charge, toner attracted, transferred, fused) earn the marks.
- Saying a monitoring system "controls" something. If nothing changes the physical process, it is monitoring; add the actuator and the feedback and it becomes control.
- Writing that RAM "stores files permanently" or that ROM "stores the user's data". RAM is volatile working memory; ROM holds the fixed start-up instructions.
- A truth table with fewer than 2^n rows, or rows in a random order. Count in binary from 000 to 111 so no combination is missed.
- Drawing two lines from one output of a gate to be safe, or leaving a wire that ends nowhere. Draw exactly the connections the expression needs.

Processor Fundamentals

A-Level Computer Science

Von Neumann architecture

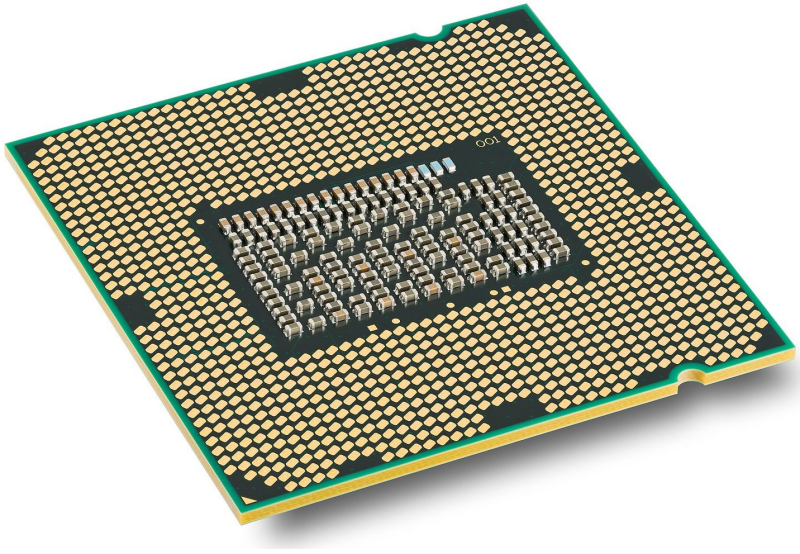
The **Von Neumann architecture** · underlies almost every general-purpose computer:

- a **single memory** – the **Immediate Access Store** (IAS) – holds both **program instructions** and **data** (the **stored program** concept).
- a **processor** (CPU) fetches instructions from memory and runs them one at a time.
- instructions run **in order** unless a branch changes the flow.

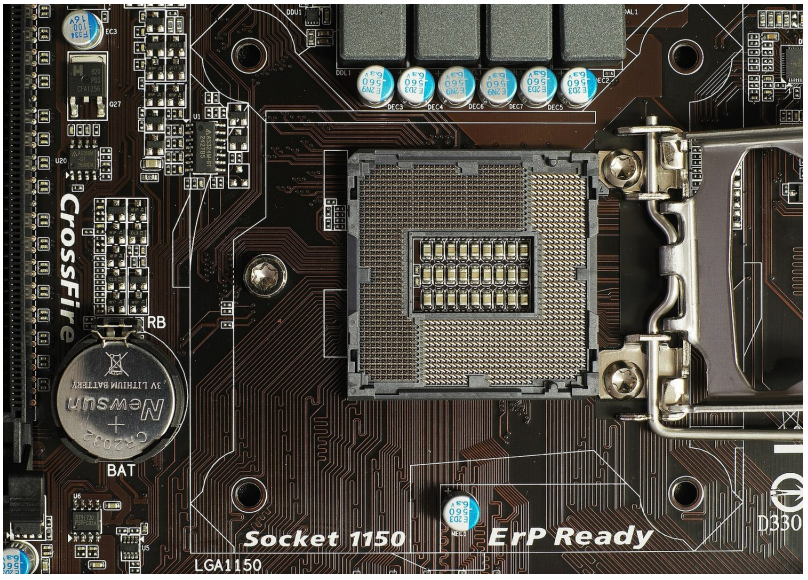
The stored-program idea is what makes a computer flexible: change the program and you change what it does, with no rewiring.

The CPU's main parts

All of these parts sit inside one small chip. The diagram later in this section shows how they connect; the photo below shows the real thing.



A modern CPU: the whole processor is one small chip (here seen from below, showing the contacts)



The matching CPU socket on the motherboard: the chip's contacts press onto these pins

■ Arithmetic and Logic Unit (ALU)

The **ALU** does the arithmetic (add, subtract, …) and logic (AND, OR, comparisons). It takes operands from **registers** and puts results back in a register.

■ Control Unit (CU)

The **control unit** **decodes** each instruction and sends the **control signals** to carry it out – opening data paths, telling the ALU what to do, and controlling memory reads and writes.

■ System clock

The clock sends a steady stream of pulses that keep the CPU in step. Each instruction takes a fixed number of cycles, and the **clock speed** (e.g. 3.8 GHz) is one factor in performance.

"Explain how the CU and the system clock work together": the clock emits pulses at a fixed frequency; the control unit uses each pulse to move the fetch-execute cycle on by one step, sending its control signals in time with the pulses, so every part of the processor changes state together. A faster clock means more steps per second, up to the point where the circuits cannot settle between pulses.

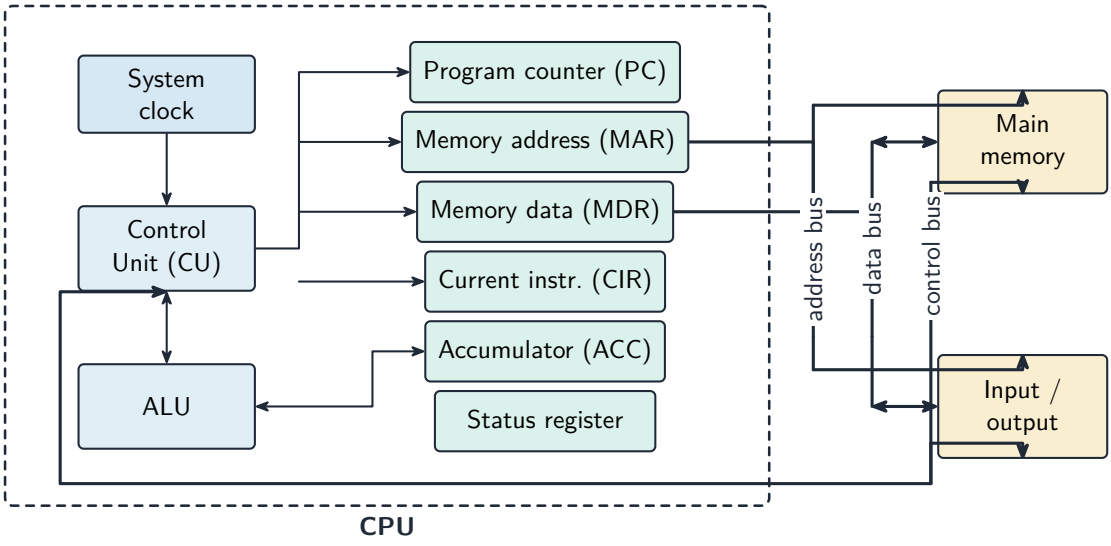
Registers

Registers are tiny, very fast stores inside the CPU. The **special purpose registers** each have a fixed job in the cycle:

- **Program Counter** (PC) – the address of the **next** instruction.
- **Memory Address Register** (MAR) – the address being read or written.
- **Memory Data Register** (MDR) – the data going to or from memory.
- **Current Instruction Register** (CIR) – the instruction being decoded.
- **Accumulator** (ACC) – the value the ALU is working on.
- **Status Register** – holds **flags** (carry, zero, negative, overflow) used by branches. Each flag is one bit, set or cleared by the ALU after an operation: the zero flag after a comparison that matched, the carry flag when an addition overflowed the register, the negative flag when a result is negative. A conditional jump reads the flags to decide whether to branch, and an overflow flag can raise an interrupt.
- **Index Register** – an offset added to an address in indexed addressing; incrementing it steps through an array one element at a time.

The "complete the table describing the role of each register" question wants one precise sentence per register in these terms: the PC holds the address of the next instruction to be fetched; the MAR holds the address of the location being read from or written to; the MDR holds the data or instruction just read from, or about to be written to, that location; the CIR holds the instruction currently being decoded and executed; the ACC holds the result of the last arithmetic or logic operation.

General-purpose registers are used by the programmer for temporary values during a calculation. Movements of data between registers and memory are written in **register transfer** notation – e.g. `MAR ← [PC]` ("copy the contents of PC into MAR").



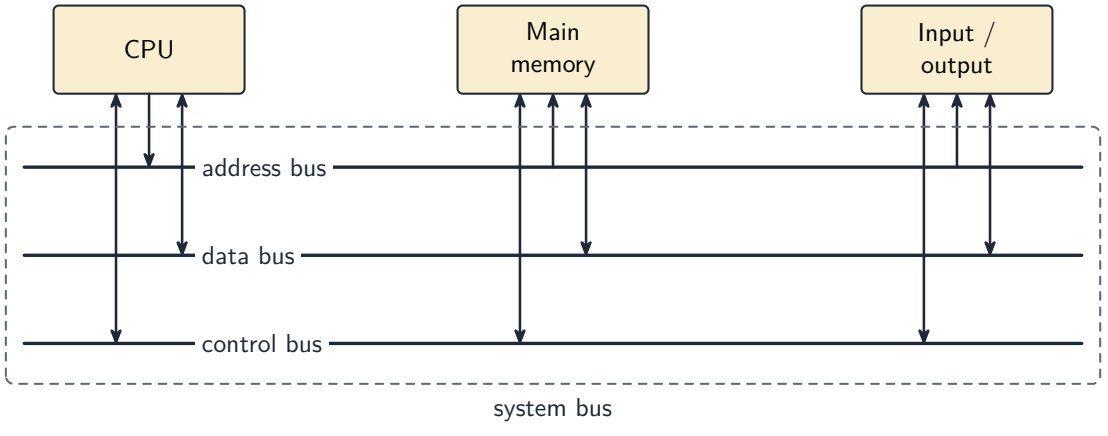
The Von Neumann CPU: registers, control unit and ALU linked by buses

Buses

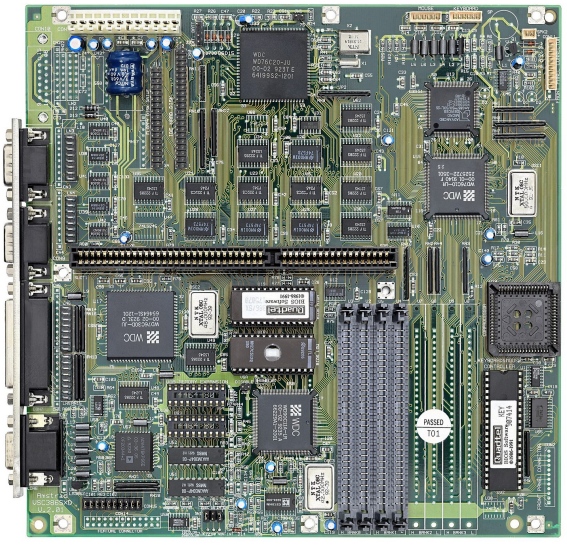
Three internal **buses** (sets of parallel wires) connect the parts:

- **address bus** – carries the memory address. **One-way** (CPU → memory).
- **data bus** – carries the data. **Two-way**.
- **control bus** – carries control signals (read, write, interrupt). **Two-way**.

An n-bit address bus can reach 2^n memory locations. The data-bus width sets how many bits move per access (often the word size).



The three system buses connecting the CPU, memory and input/output



A motherboard: the CPU, memory and I/O all sit on one set of buses – the printed tracks running between them

What affects performance

- **clock speed** – more cycles per second.
- **number of cores** – a multi-core CPU runs several threads at once.
- **word size** – a 64-bit CPU handles 64-bit chunks per cycle and can address far more memory than a 32-bit one.
- **amount of RAM** – more RAM holds more of the working set; too little forces the OS to **page** to disk.
- **cache memory** size – more cache cuts average memory access time.
- **secondary storage** type – an SSD loads programs far faster than an HDD.
- **bus width and speed** – wider/faster buses move data more quickly.

Match the specs to the workload: a quad-core beats a dual-core on parallel work, but higher per-core speed wins on single-threaded work.

Each factor is a two-mark answer with a reason attached:

- **More cores**: each core can fetch and execute its own instruction at the same time, so several programs, or the threads of one program, run in parallel. But a program must be written to use more than one core, so doubling the cores does not double the speed.
- **Higher clock speed**: more fetch-execute cycles per second, so more instructions per second; the limit is the heat produced.

- **Wider bus**: a wider data bus moves more bits in each transfer, so fewer transfers are needed for the same data; a wider address bus can address more memory locations.
- **Cache memory**: a small, fast memory inside or next to the processor that keeps the instructions and data used most recently or most often. Reading them from cache is much faster than from RAM, so the processor spends less time waiting.

"Explain why the new computer performs better" is answered by comparing the two specifications line by line: a higher clock speed executes more instructions per second, more cores run more tasks at once, more cache means fewer slow accesses to RAM, and more RAM means fewer transfers to disk.

Ports

A **port** is a physical socket for connecting a **peripheral** :

- **USB** (Universal Serial Bus) – general-purpose (keyboards, drives, phones).
- **HDMI** (High Definition Multimedia Interface) – digital video and audio to a screen.
- **VGA** (Video Graphics Array) – older analogue video output to a monitor.
- **Ethernet (RJ-45)** – wired LAN. Audio jacks – headphones/microphone.

Different ports use different signals, so an HDMI cable will not fit a USB socket. USB-C is unusual in carrying video, data and power.

"Explain how the computer connects to the monitor through HDMI": the HDMI port sends the video and the audio as one digital signal down a single cable, so no conversion to analogue is needed and the picture is not degraded; the cable carries high-definition resolutions and the monitor's own port decodes the signal. A USB device is **plug-and-play**: when it is connected the computer detects it, identifies it, loads or installs the driver it needs, and can supply it with power, all without a restart.

Fetch-Execute cycle

The CPU repeats the **fetch-execute cycle** - , one run per machine instruction.

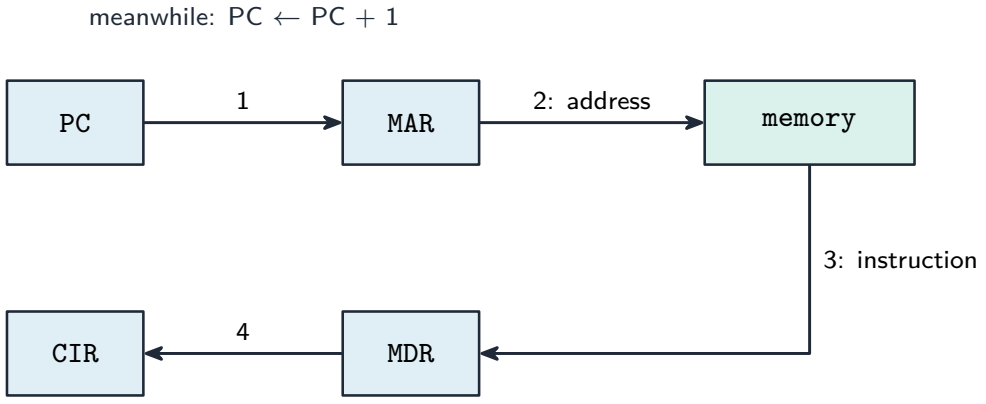
Fetch

1. the PC's address is copied to the MAR.
2. the PC is **incremented** to point to the next instruction.
3. a **read** signal goes over the control bus.
4. memory puts the instruction on the data bus.
5. it is copied into the MDR, then into the CIR.

The exam asks for these steps in **register transfer notation** , where `[X]` means the contents of register X and `[[MAR]]` means the contents of the memory location whose address is in the MAR:

```
MAR ← [PC]          the address of the next instruction goes to the MAR
PC  ← [PC] + 1      the PC now points to the following instruction
MDR ← [[MAR]]       the instruction at that address is read into the MDR
CIR ← [MDR]         the instruction is copied into the CIR for decoding
```

The order matters: the PC is incremented straight after its address has been copied, so that a jump executed later can still overwrite it. During execution the same notation describes each instruction; for `LDD 200`, for example, `MAR ← 200`, `MDR ← [[MAR]]`, `ACC ← [MDR]`.



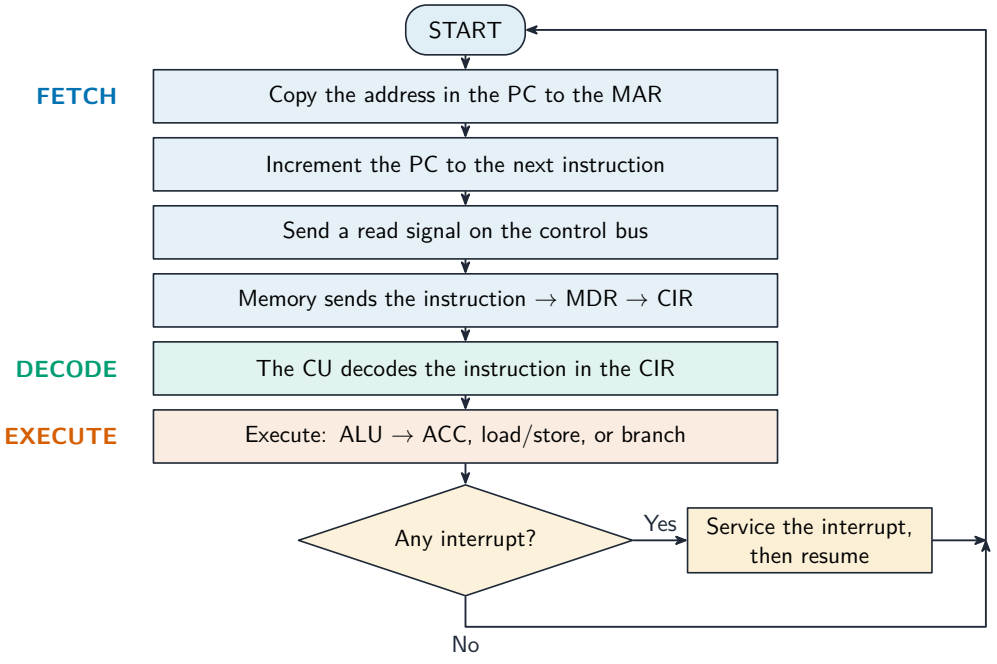
The register transfers in a fetch: PC → MAR → memory → MDR → CIR, with the PC incremented

Decode

The CU decodes the instruction in the CIR – what operation, and which operands or addresses.

Execute

The CU carries it out: arithmetic/logic goes to the ALU (result to the ACC); a load/store moves data between memory and a register; a branch changes the PC. Then the cycle repeats.



The fetch-execute cycle, with a check for interrupts each time

Interrupts

An **interrupt** is a signal that **pauses** the normal cycle so the CPU can handle an urgent event (a key press, a packet arriving, a hardware fault, division by zero, the OS timer).

Handling one:

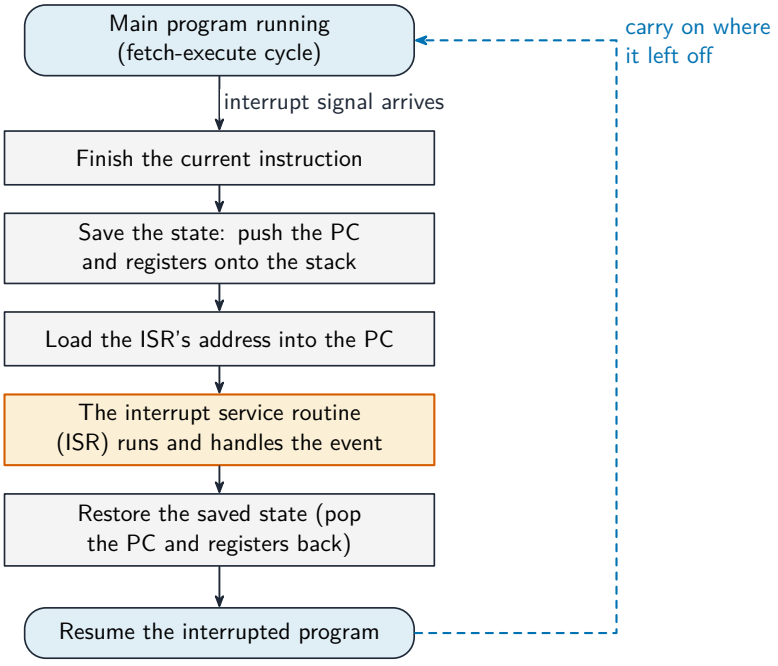
1. finish the current instruction.
2. **save the state** (PC and registers).
3. load the address of the **interrupt service routine** (ISR) into the PC and run it.
4. the ISR handles the event.
5. **restore** the saved state and carry on.

Interrupts let the system respond promptly without the CPU constantly checking devices, and are how the OS multitasks.

"Explain how an interrupt from an input device is detected and handled in the F-E cycle" is a four-mark answer with these points: the device sends an interrupt signal that sets the interrupt flag in the **interrupt register** ; the processor checks that register at the **end of every fetch-execute cycle**, after the current instruction has finished executing; if a flag is set and the interrupt has a higher priority than the current task, the contents of the PC and the other registers are saved onto the **stack** ; the address of the interrupt service routine is loaded into the PC and the routine

runs; when it finishes, the saved values are restored from the stack and the interrupted program continues from where it stopped.

Causes worth naming: a hardware interrupt from a device (a key pressed, a printer buffer empty, a network packet arriving), a software interrupt from a fault (division by zero, an illegal instruction, arithmetic overflow), a timer interrupt from the operating system marking the end of a time slice, and a power failure warning.



How an interrupt fits into the fetch-execute cycle

Assembly language and machine code

The CPU actually runs **machine code** – bit patterns, specific to one architecture. **Assembly language** is a readable form, with one instruction per machine instruction, written using **mnemonics** like `LDD`, `ADD`, `JMP`. An **assembler** translates it to machine code.



An assembler turns mnemonics into machine-code bit patterns

Two-pass assembler

A two-pass assembler reads the source twice:

- **pass 1** builds a **symbol table** : each time a **label** (like `LOOP:`) appears, record its address; no code yet.
- **pass 2** generates code: translate each instruction, and when one refers to a label (like `JMP LOOP`), look up its address in the symbol table.

Two passes handle **forward references** (a jump to a label defined later).

Worked example. Apply the two-pass process to this program, whose first instruction is stored at address 100.

```
        LDD  COUNT
LOOP:   DEC  ACC
        CMP  #0
        JPN  LOOP
        END
COUNT:  5
```

Pass 1 reads each line, counts the address it will occupy, and records every label in the symbol table: `LOOP` = 101 (the `DEC` line) and `COUNT` = 105 (the data line). No code is produced. Pass 2 reads the program again and translates each line into machine code, replacing each mnemonic by its **opcode** and each symbolic address by the number from the symbol table: `LDD COUNT` becomes the opcode for `LDD` with **operand** 105, and `JPN LOOP` becomes the opcode for `JPN` with operand 101. The jump back to `LOOP` could have been resolved in one pass, but a jump *forward* to a label not yet seen could not, which is why the assembler makes two.

Example instruction set

Cambridge uses a small generic set, printed in the paper's reference table, with one general-purpose register, the accumulator (ACC), and an index register (IX). An operand written `#n` is a denary number, `Bn` a binary number and `&n` a hexadecimal number; `<address>` is a location number or a label.

Group	Instruction	What it does
Data movement	`LDM #n`	load the number n into ACC (immediate)
	`LDD <address>`	load the contents of the address into ACC (direct)
	`LDI <address>`	the address holds another address; load the contents of that one into ACC (indirect)

Group	Instruction	What it does
	`LDX <address>`	add IX to the address and load the contents of the result into ACC (indexed)
	`LDR #n`	load the number n into IX
	`MOV <register>`	copy ACC into the named register (IX)
	`STO <address>`	store the contents of ACC at the address
Input and output	`IN`	read a key press and put its ASCII code in ACC
	`OUT`	output the character whose ASCII code is in ACC
Arithmetic	`ADD <address>` / `ADD #n`	add the contents of the address, or the number, to ACC
	`SUB <address>` / `SUB #n`	subtract from ACC
	`INC <register>` / `DEC <register>`	add 1 to, or subtract 1 from, ACC or IX
Compare	`CMP <address>` / `CMP #n`	compare ACC with the contents of the address, or with n, and set the flag
	`CMI <address>`	compare ACC with the contents of the address held at the address (indirect)
Jump	`JMP <address>`	jump to the address unconditionally
	`JPE <address>` / `JPN <address>`	jump if the last compare was equal / not equal
Bit manipulation	`AND`, `OR`, `XOR` with `#n`, `Bn`, `&n` or `<address>`	bitwise operation on ACC
	`LSL #n` / `LSR #n`	shift ACC logically n places left or right
	`END`	end the program

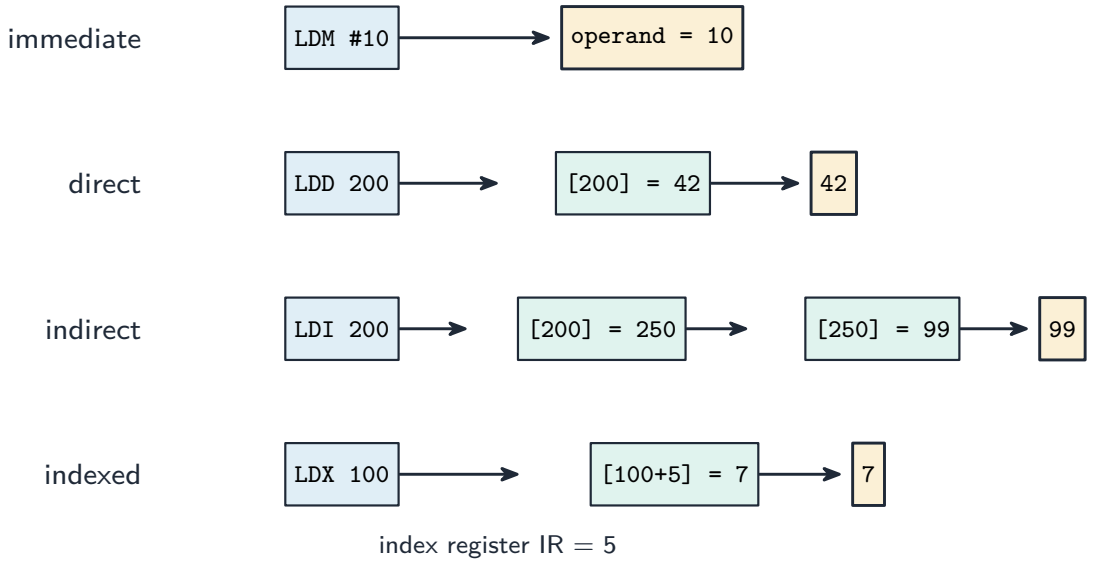
The "assembly language instructions are grouped" question wants the group names, and an instruction from each: data movement, input and output, arithmetic, unconditional and conditional jumps, compare, and bit manipulation.

Addressing modes

The **addressing mode** (the **modes of addressing**) says how the CPU finds the operand:

- **immediate addressing** – the operand is the value in the instruction. `LDM #10` loads 10.
- **direct addressing** – the instruction holds an address; the operand is the value there. `LDD 200`.
- **indirect addressing** – the instruction holds an address that holds **another** address, which is the data. `LDI 200`.
- **indexed addressing** – effective address is `address + index register`; used for arrays. `LDX 100` with IR = 5 reads address 105.

(**Relative addressing** gives the address as an offset from the PC – used for jumps.)



How each addressing mode reaches its operand – immediate, direct, indirect and indexed

Worked example. Memory holds: location `200` = `250`, location `250` = `99`, location `105` = `7`. The index register holds `5`. What is in the accumulator after each of `LDM #200`, `LDD 200`, `LDI 200` and `LDX 100`? Follow how far each mode has to look. `LDM #200` is **immediate** - the operand *is* the number written in the instruction, so the accumulator holds **200**. `LDD 200` is **direct** - go to location 200 and take what is there: **250**. `LDI 200` is **indirect** - location 200 holds 250, which is *another address*, so go on to location 250: **99**. `LDX 100` is **indexed** - add the index register to the address, 100 + 5 = 105, and read location 105: **7**. Count the hops to keep them apart: immediate 0, direct 1, indirect 2, indexed 1 (once the index has been added).

Tracing an assembly program

To **trace** it: make a table with columns for the PC, ACC, index register, each variable and any flags. Step through the instructions, updating the table after each; follow branches when they change the PC; stop at `END`. A common pattern is a loop over an array using indexed addressing.

Worked example. Trace this program. Address 200 holds 5 and address 201 holds 0.

```
100   LDD  200
101   CMP  #0
102   JPE  108
103   OUT
104   DEC  ACC
105   STO  200
106   LDD  201
107   JMP  100
108   END
```

Write one row for each instruction executed, filling in only the columns that change:

Instruction	ACC	200	201	Output
start		5	0	
`LDD 200`	5			
`CMP #0`				
`JPE 108`				not taken
`OUT`				character with code 5
`DEC ACC`	4			
`STO 200`		4		
`LDD 201`	0			
`JMP 100`				
`LDD 200`	4			

and so on, until `LDD 200` loads 0, the compare sets the equal flag, `JPE 108` is taken and the program ends. Three things the examiner checks: a `CMP` changes no register, only a flag; a jump not taken still counts as executed; and `OUT` outputs a character, so it goes in the output column, not the ACC column. "State the effect of changing `LDD 10` to `LDM #10`": the ACC would hold the number 10 instead of the contents of address 10.

Binary shifts

A **logical shift** moves all the bits left or right by some places, filling new positions with 0.

- **left shift by 1** (`LSL #1`) – bits move left, a 0 enters on the right; for an unsigned number this is $\times$ **2**.
- **right shift by 1** (`LSR #1`) – bits move right, a 0 enters on the left; for an unsigned number this is **integer $\div$ 2**.

Shifting by n places multiplies or divides by 2^n. Example: `00001011` (11) `LSL #1` $\rightarrow$ `00010110` (22).

Bits shifted off the end are lost, so the multiplication is only correct while they were zeros. `LSL #2` on the two's-complement integer `11001010` gives `00101000`: the two 1s that fell off the left are gone, the sign bit has changed, and the result is no longer four times the original.

An **arithmetic right shift** keeps the sign bit so a negative signed number stays negative. A **cyclic shift** (rotate) feeds the bit that drops off one end back in at the other end, so no bits are lost.

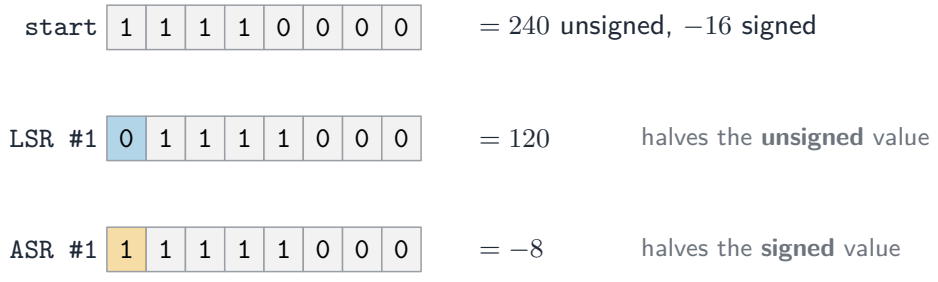
"Show the result of an arithmetic right shift of 3 places on `10011110`": copy the sign bit into each vacated place, `11110011`. The same shift on `01011100` gives `00001011`. A cyclic left shift of 1 on `10000110` gives `00001101`: the leading 1 reappears on the right.

LSL #1	00001011	$\longrightarrow$	00010110	$\times 2$	(0 in on the right)
LSR #1	00001011	$\longrightarrow$	00000101	$\div 2$	(0 in on the left)
ASR #1	10110100	$\longrightarrow$	11011010	sign bit kept (stays negative)	

Logical left ($\times 2$), logical right ($\div 2$) and arithmetic right (keeps the sign bit)

The difference between the two right shifts is a single bit. Take `11110000`, which is 240 read as unsigned and -16 read as signed. `LSR #1` brings in a 0 and gives `01111000` = 120, which is the correct half of 240. `ASR #1` copies the sign bit instead and gives

`11111000` = −8, which is the correct half of −16. Neither is wrong – each halves the value under one reading.



Logical and arithmetic right shift on the same byte: only the bit that enters on the left differs

■ Bit manipulation for monitoring/control

Embedded devices often use one **bit** of a register per signal (e.g. bit n = LED n). Using a **mask** – **bit masking** – you can:

- **set** bit n: `R = R OR` a mask with bit n set.
- **clear** bit n: `R = R AND` a mask with bit n clear and the rest set.
- **toggle** bit n: `R = R XOR` a mask with bit n set.
- **test** bit n: `R AND` the mask, then check if the result is non-zero.

set bit 2		**clear bit 6**		**toggle bit 3**	
	01001000		01001000		01001000
OR	00000100	AND	10111111	XOR	00001000
=	01001100	=	00001000	=	01000000

Set a bit with OR, clear it with AND, toggle it with XOR – each using a mask

Bit manipulation is fast, uses little memory, and lets one byte hold up to 8 on/off states.

In the exam's instruction set these are `AND`, `OR` and `XOR` with a mask written as a denary, binary or hexadecimal operand. With the ACC holding `10101100`:

Instruction	Mask	Result in ACC	Effect
`AND B00001111`	00001111	00001100	keeps only the low four bits (clears the others)
`OR #1`	00000001	10101101	sets the least significant bit, leaving the rest unchanged
`XOR &FF`	11111111	01010011	inverts every bit
`AND B00001000` then `CMP #0`	00001000	00001000	tests bit 3: the compare is not equal, so bit 3 was set
`LSL #2`		10110000	shifts left two places, losing the top two bits
`LSR #3`		00010101	shifts right three places, zeros entering on the left

"Write the instruction that sets the least significant bit to 1 and leaves the others unchanged": `OR #1`, or `OR B00000001`. To clear a bit use `AND` with a mask that has a 0 in that place and 1s elsewhere; to test a bit, `AND` with a mask that has a 1 only in that place, then compare the result with zero. In a monitoring device, one bit of a register per sensor lets a single `AND` check whether a particular sensor is on, and one `OR` switches an actuator's control bit on without disturbing the others.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
stored program concept	the program instructions and the data are both held in main memory, and instructions are fetched and executed one at a time
register	a small, very fast storage location inside the processor with a specific purpose
Program Counter	the register holding the address of the next instruction to be fetched
Memory Address Register	the register holding the address of the memory location being read from or written to
Memory Data Register	the register holding the data or instruction just read from, or about to be written to, memory

Term	Definition
Current Instruction Register	the register holding the instruction currently being decoded and executed
Accumulator	the general-purpose register holding the result of the last arithmetic or logic operation
cache memory	small, fast memory close to the processor holding frequently used instructions and data
interrupt	a signal from a device or program that causes the processor to pause the current task and run an interrupt service routine
assembly language	a low-level language in which each mnemonic instruction corresponds to one machine-code instruction
immediate addressing	the operand is the value written in the instruction
direct addressing	the operand is the contents of the address written in the instruction
indirect addressing	the address in the instruction holds the address of the operand
indexed addressing	the operand's address is the address in the instruction plus the contents of the index register
relative addressing	the operand's address is given as an offset from the address of the current instruction
logical shift	every bit moves the given number of places and zeros fill the vacated places

Exam tips

- Learn the **fetch-execute cycle** in register-transfer terms (PC, MAR, MDR, CIR, ACC) and what increments the PC.
- Name each register's job; the **address bus is one-way**, the data bus is two-way.
- Distinguish the **addressing modes** (immediate, direct, indirect, indexed) – a frequent question.
- Explain how clock speed, number of cores, cache size and word length affect **performance**.
- For a **binary shift**, state whether it is logical or arithmetic; a left shift multiplies by 2, a right shift divides by 2.

Common mistakes

- Saying the PC holds the current instruction, or the MDR holds an address. The PC holds the address of the *next* instruction; the MDR holds data or an instruction, never an address.
- Leaving the increment of the PC out of the fetch, or putting it after the execute. It happens as soon as the address has been copied to the MAR.
- Reading `LDD 10` as "load 10". `LDD 10` loads the contents of address 10; `LDM #10` loads the number 10.
- Putting a value in the ACC column for `CMP` or `OUT`. A compare sets a flag only; an output goes to the output column.
- Saying an interrupt is handled "immediately". The processor finishes the current instruction and checks for interrupts at the end of the cycle.
- Using a logical right shift on a negative two's-complement number. Only an arithmetic shift keeps the sign bit.

System Software

A-Level Computer Science

Operating systems

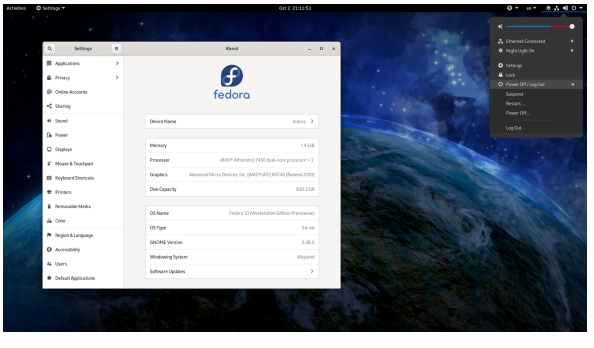
Why a computer needs an OS

Hardware on its own can only fetch and run instructions – it knows nothing about files, programs, networks or users. The **operating system** (OS) is the **software layer** that:

- **manages the hardware** (**processor**, memory, I/O, storage) for the running programs.
- **provides services** (file system, network, user accounts) through a clear interface, so programs need not talk to the hardware directly.
- **provides a user interface** (command line, GUI, touch).
- **lets several programs share the hardware safely** – each gets fair CPU time and is kept out of the others' memory.

Without an OS, every program would need its own drivers, and only one program could safely run at a time.

"Describe the purpose of an OS" – the five-mark list. The OS (1) provides an **interface** between the user and the hardware; (2) **hides the complexity** of the hardware from the user and from application programs; (3) **manages the hardware resources** – processor time, memory, storage and input/output devices – and shares them between programs; (4) loads application software into memory and runs it, giving every program the same platform to run on; (5) lets several programs run at once (**multitasking**) while keeping them, and the users' data, secure. Give five *different* points; "it runs the computer" or "it manages resources" alone earns nothing.



A desktop operating system manages the screen, files and programs for the user

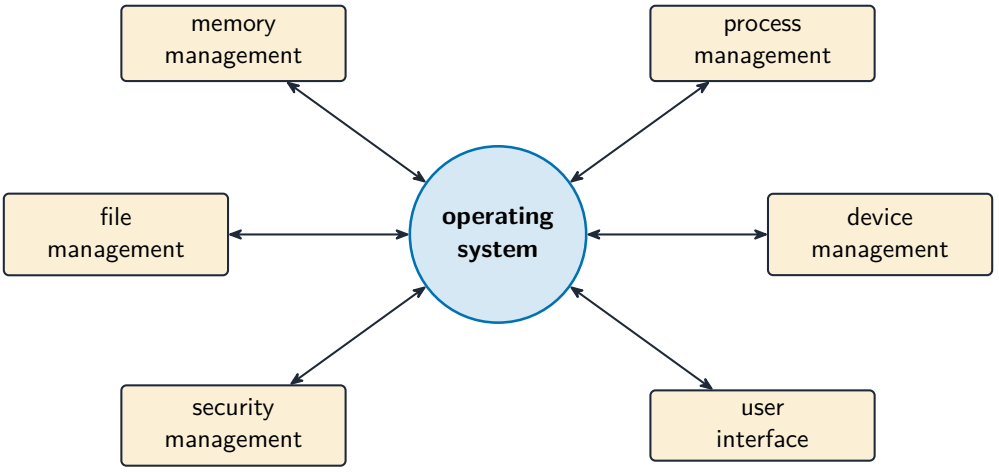


A phone needs one just as much: this is a mobile operating system, Android

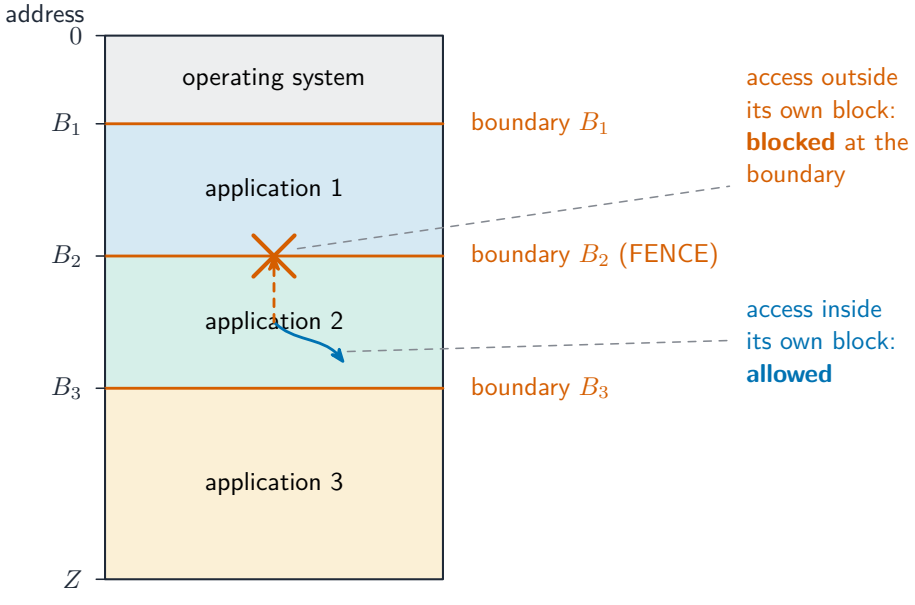
Key management tasks

The syllabus names five. Each point below is one thing the OS actually does, which is what a "describe" question wants.

- **memory management** – allocates memory to each program when it is loaded, keeps every program's memory separate (**memory protection**) so one cannot overwrite another, frees the memory when a program ends, and swaps pages between **RAM** and **secondary storage** (the disk) (**virtual memory** / **paging**) so more programs can be open than physical memory allows.
- **process management** – a running program is a **process**. The OS creates and ends processes, decides which process gets the CPU next (**scheduling**) and for how long (a **time slice**), switches between them, resolves conflicts when two want the same resource, and can kill one that stops responding.
- **hardware management** (input/output and peripherals) – talks to each device through its **device driver**, queues and buffers data going to slow devices such as a printer, responds to **interrupts** from devices, and shares one device between several programs.
- **file management** – creates, names, copies, moves and deletes files and folders, keeps the **directory** structure and a record of where each file is stored on the disk, allocates disk space to files, and enforces **access rights** (read / write / execute) for each user.
- **security management** – user accounts and passwords (**authentication**), access rights, encryption of stored data, a firewall, automatic security updates, and a log of who did what.



The main jobs the operating system manages



Memory protection keeps each application in its own block of memory

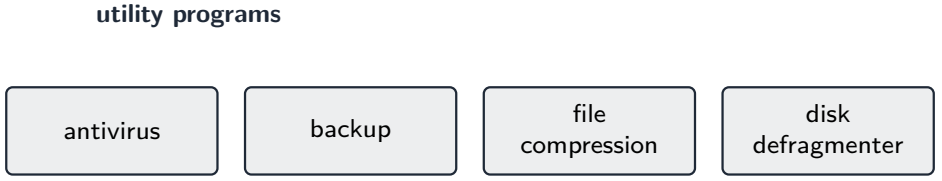
How memory and process management support multitasking (a four-mark favourite). Memory management loads several programs into memory at the same time, each in its own protected area, and keeps track of which addresses belong to which; process management shares the processor between them – each process runs for a time slice, the OS saves its state and switches to the next, and the switching is so fast that all the programs appear to run together. Interrupts let the OS take the processor back from a process whenever a device needs attention.

Interrupts. A **hardware interrupt** comes from a device: a key pressed, a mouse click, a printer out of paper, a disk finishing a transfer, a power failure. A

software interrupt comes from a program: division by zero, an invalid instruction, an attempt to use memory it does not own, or a request for an OS service. The OS's **interrupt handler** saves the state of the running process, deals with the interrupt, then restores the process (topic 4 covers the fetch–execute detail).

■ Utility software

Utility programs are system software that maintain, repair or optimise the computer rather than doing a user's task; the examiner accepts "performs a specific maintenance task that improves performance or security". Most OSes bundle these:



Utility programs: antivirus, backup, compression and defragmenter

- **disk formatter** – prepares a new disk (or wipes an old one) for use: sets up its **file system** and partitions, deleting any existing data.
- **virus checker** (**antivirus**) – scans files and memory, compares code against a database of known virus **signatures** and watches for suspicious behaviour, then quarantines or deletes what it finds; runs on a schedule and on every download, and needs updating as new viruses appear.
- **defragmentation software** (**disk defragmenter**) – a hard disk stores a file in whatever free blocks it finds, so after many saves and deletes a file is scattered (**fragmented**) across the platter and the read/write head must jump between the pieces. The defragmenter moves the pieces of each file next to each other and gathers the free space into one region, so files load faster and new files are not fragmented. Not needed on an SSD, which has no moving head.
- **disk contents analysis / disk repair software** – shows what is using the disk space (large, duplicate or temporary files) so they can be removed; finds and repairs bad sectors, lost clusters and file-system errors.
- **file compression** (**compression**) – shrinks files so they need less storage and transfer faster; archiving bundles many files into one.
- **back-up software** (**backup**) – copies files to another medium (external disk, network, cloud) on a schedule so data can be restored after loss, corruption or a ransomware attack; a full copy is followed by **incremental backups** of only what changed.
- a **firewall** (filters network traffic by rules) and encryption tools, for security; a **system monitor** and automatic **updates**.

Bundling these with the OS saves the user installing each one.

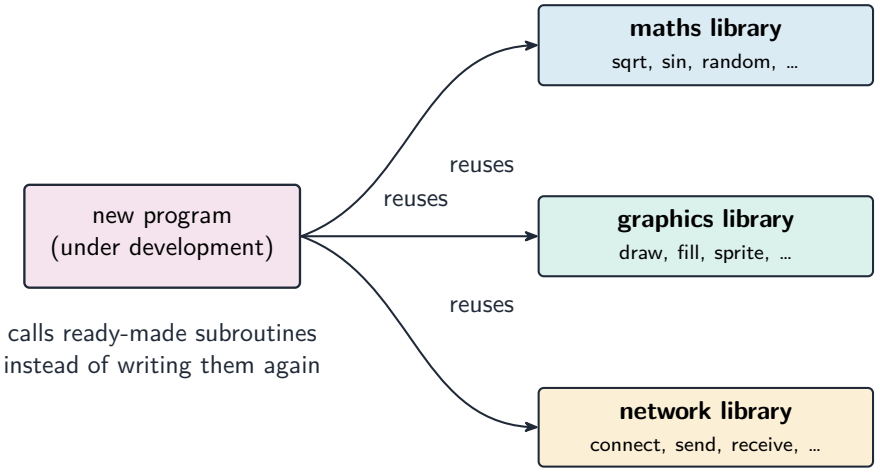
Which utility does what. *Performance:* defragmentation (faster file access), disk repair (a disk with errors is slow or fails), disk contents analysis (free space by deleting junk), compression (more fits on the disk). *Security:* virus checker, firewall, encryption, and backup (the only recovery from ransomware). A "draw one line" question pairs each utility with exactly one purpose – learn the pairs above and use the syllabus names.

> **Worked example.** Explain how defragmentation can improve the performance of a computer (3 marks).

Over time a file is stored in blocks scattered across the hard disk, so reading it needs many movements of the read/write head. The defragmenter rearranges the blocks so each file is stored **contiguously** and the free space is together. Files are then read with fewer head movements, so they load faster, and new files can be written into one continuous space.

Program libraries

A **program library** is pre-written code (**subroutines**, classes, modules) that programs reuse instead of writing it themselves – e.g. a maths library, a network library, a graphics library.



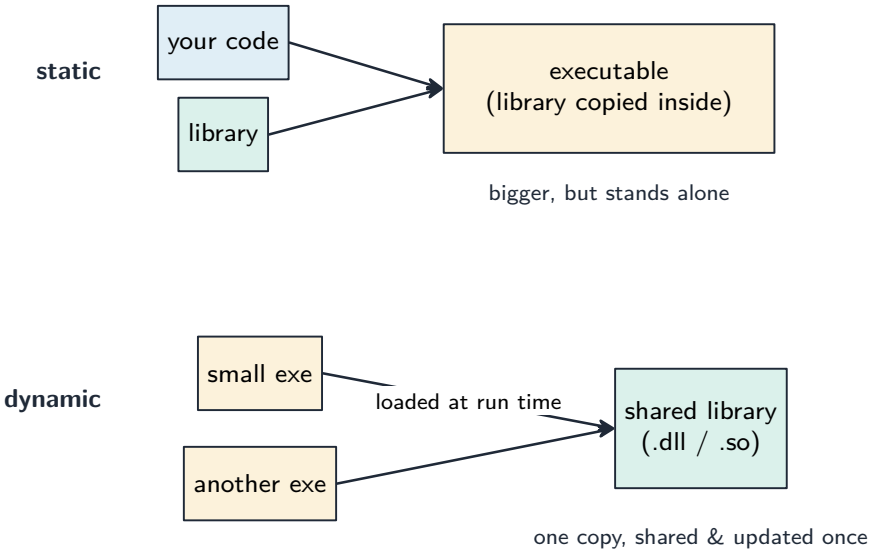
A new program reusing ready-made routines from libraries

Benefits: **saves time** (off-the-shelf code), **reliable** (well-tested, widely used), and **standardised** (consistent behaviour).

The examiner's benefit list, for the developer. The **library routines** are already written and tested, so development is faster and cheaper; they are reliable and, being used by many programs, largely error-free; the developer needs no expertise in that area (graphics, compression, encryption, path-finding); the program is easier

to maintain because common code lives in one place; and a whole team can use the same routines, giving consistent results. **Drawbacks:** a routine may not do exactly what you need and you cannot change it; your program depends on the library being available, correct and secure – a bug or a security hole in the library is a bug in your program; and you must learn how to call it.

- a **static library** is copied into the executable at compile time (stands alone, but larger and needs rebuilding to update).
- a **dynamic library** (DLL, Dynamic Link Library; `.so`) is loaded at run time (smaller executables, shared by many programs, updated once for all).



Static: the library is copied into the executable. Dynamic: a shared library file is loaded at run time

Dynamic Link Library (DLL) files. A DLL is a library that is loaded into memory only when a program calls it, at **run time**, and stays as a separate file rather than being copied into the executable. *Benefits:* the executable is smaller; several running programs share one copy of the DLL in memory; a DLL can be updated (bug fix, new device) without recompiling the programs that use it; and memory is used only while the routine is needed. *Drawbacks:* the program will not run if the DLL is missing, moved or the wrong version; an updated DLL can break a program that relied on the old behaviour; and a fake DLL put in its place runs with the program's rights.

> **Worked example.** A team writing the software for a restaurant robot uses a program library that includes a routine to find the shortest path between tables. Explain two benefits and one drawback to the team.

Benefits: the routine is already written and tested, so the team saves time and can trust the result; the team need not understand path-finding algorithms themselves and

can spend the time on the robot's own features. *Drawback:* the routine may not handle the restaurant's exact needs (moving chairs, one-way aisles) and the team cannot alter it, so they may have to work around its limits.

Language translators

You write source code; the computer runs **machine code** . A **translator** converts between them.

■ Assembler

An **assembler** translates **assembly language** into machine code: each mnemonic instruction (`LDD`, `ADD`, `JMP`) becomes exactly one machine-code instruction, and symbolic addresses and labels are replaced by real addresses. It is needed because the processor executes only machine code, and assembly is used where the programmer needs direct control of the hardware (embedded systems, device drivers).

■ Compiler

A **compiler** translates a high-level program into machine code **once, before it runs**.

- it reports **all** errors at compile time; once clean, it produces a stand-alone **executable** that runs **without the compiler installed** and can be run many times.
- generally **faster at run time** (no translation while running), but tied to one CPU/OS – recompile for each platform.

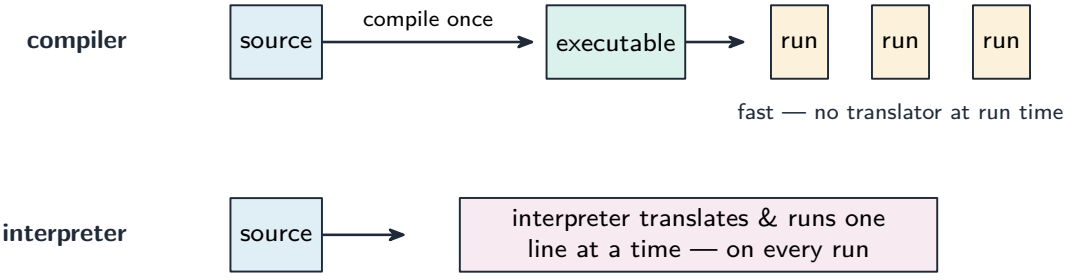
The two-mark description: *a compiler translates the **whole** high-level program into machine code (**object code**) **before** it is run, produces an executable file, and reports **all** the syntax errors together as a list at the end of translation.* It does not run the program.

■ Interpreter

An **interpreter** translates and runs a high-level program **one line at a time**, producing no executable.

- it reports an error **when it reaches that line**, then stops; you can fix it and continue – good for development.
- the **interpreter must be installed** to run the program; generally **slower** (each run re-translates), but easy to **port** across platforms.

The two-mark description: *an interpreter translates **one statement** of the high-level program at a time and **executes** it immediately before moving to the next; no executable file is produced; it stops at the **first** error it meets and reports it.* Both the source code and the interpreter must be present every time the program runs.



A compiler translates once into a standalone program; an interpreter translates line by line, every run

Choosing between them

Use a compiler when:	Use an interpreter when:
run-time speed matters	you want fast edit–run cycles
distributing to users without dev tools	writing cross-platform scripts
the program runs many times	the program is small or run once
	teaching beginners

Benefits and drawbacks, as the mark scheme lists them.

	compiler	interpreter
execution speed	fast – already machine code	slower – translated on every run
what the user needs	only the executable; no translator, and the source code stays private	the source code and the interpreter
finding errors	all errors listed at once, after the whole program is translated	each error reported at the line where it occurs, as you develop
changing the code	recompile the whole program after every change	edit and run again immediately
portability	machine code runs on one platform only; recompile for each	the same source runs wherever an interpreter exists

> **Worked example.** A developer uses an interpreter while writing a program and a compiler when it is finished. Explain how each is used (4 marks).

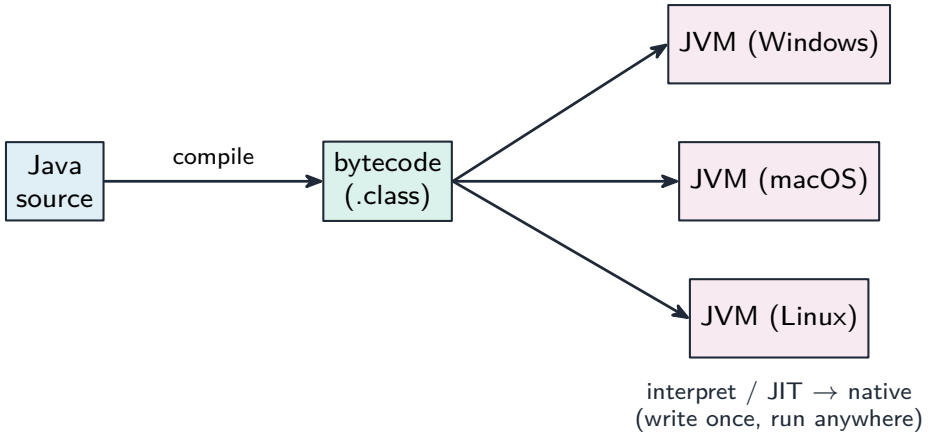
During development the interpreter runs the partly written program at once, without waiting for a complete translation; when it meets an error it reports the line, so the developer fixes it and runs again immediately – a fast edit–run cycle that is easier for

debugging. When the program is finished, the compiler translates the whole program into an executable that runs faster, needs no translator on the user's computer, and does not reveal the source code, so it can be sold to the public.

■ Hybrid: Java

Java is compiled into **bytecode** (a platform-independent intermediate form), which a **virtual machine** (the JVM) then interprets – or uses **just-in-time compilation** to turn hot parts into native code. So errors are caught early, the bytecode runs anywhere with a JVM ("write once, run anywhere"), and long-running programs reach near-native speed. C# and Python use similar designs.

The syllabus phrase is "**partially compiled and partially interpreted**": the compiler stage catches syntax errors and produces compact, **portable** bytecode; the interpreting stage lets that one bytecode file run on any machine that has a virtual machine, at the cost of some speed. Java in **console mode** (a text program run from the command line) is the syllabus's example.

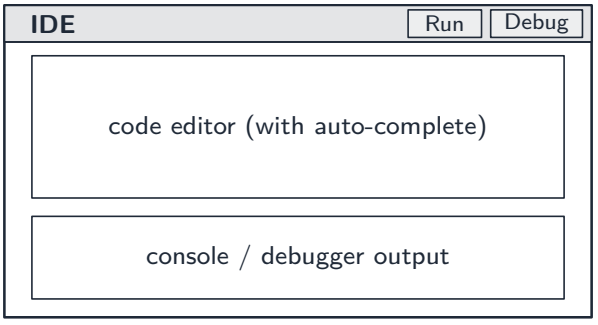


Java compiles to portable bytecode that any JVM runs – write once, run anywhere

Worked example. Java source is compiled to **bytecode**, which a JVM then interprets. Why use both, instead of compiling straight to machine code? A compiler produces machine code for **one** processor and operating system, so a program compiled on one machine will not run on another. Java's compiler instead targets a **virtual** machine, so the bytecode it produces is identical everywhere; each platform then supplies its own **JVM** to interpret that bytecode into its own native instructions. One compiled file therefore runs anywhere a JVM exists - "write once, run anywhere". The price is speed: interpreting bytecode is slower than running native code, which is why a real JVM also uses **JIT** compilation to turn frequently-run bytecode into native code while the program runs. Name **both** sides - the marks are for portability **bought at the cost of speed**.

Integrated Development Environment (IDE)

An **integrated development environment** (IDE) brings the tools to write, test and debug code into one application:

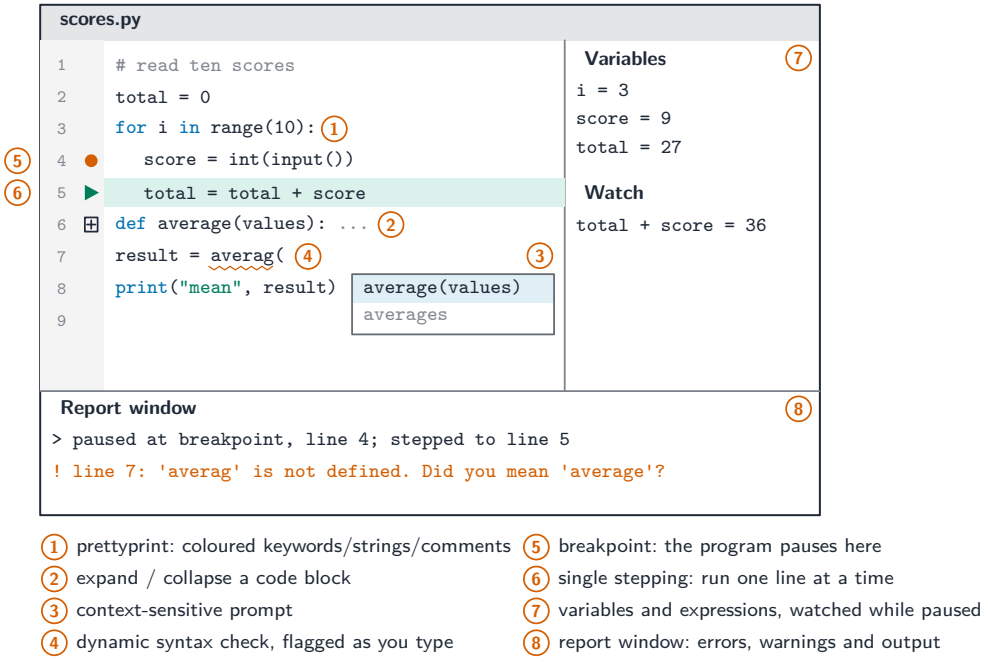


An IDE bundles the editor, a Run button and a debugger

The syllabus groups the features into four kinds. Learn which feature belongs to which, because questions ask you to sort them and to describe one from each group.

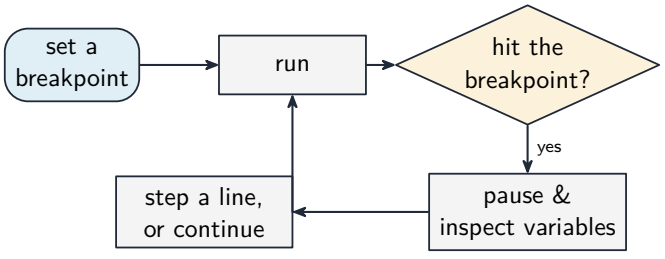
- **For coding: context-sensitive prompts** – as you type, the IDE pops up the identifiers, keywords or parameters that fit at that point in the code; **auto-complete** finishes the name for you; automatic indentation and bracket matching keep the layout right as you type.
- **For initial error detection: dynamic syntax checks** – the editor checks the syntax as you type and underlines or highlights a mistake immediately, before the program is translated; after translation, error messages with line numbers.
- **For presentation: prettyprint** – keywords, identifiers, strings and comments shown in different colours or fonts (**syntax highlighting**) with consistent indentation, so the structure is visible at a glance; **expand and collapse code blocks** – hide the body of a loop, an IF or a subroutine so you see the outline.
- **For debugging: breakpoints** – the program pauses when it reaches a marked line; **single stepping** – from the pause, run one line at a time; a window that shows the current values of **variables and expressions** as they change; and a **report window** that lists errors, warnings and output.

Other features: translator integration (compile or run with one key, errors shown inline), a **debugger** that drives the debugging features above, **version control** integration (git), project management, a help system, **refactoring** tools (safe renaming) and **unit test** integration.



The IDE features the syllabus names, in the places you would see them on screen

An IDE speeds development by putting writing → running → debugging → fixing behind one interface. Common IDEs: Visual Studio, PyCharm, Eclipse, VS Code.



A debugger: set a breakpoint, run, then pause to inspect variables and step through the code

> **Worked example.** A function `Calculate()` returns an unexpected value when the program runs. Describe how the debugging features of a typical IDE help find the cause (4 marks).

Set a **breakpoint** on the first line of `Calculate()`, so the program pauses there instead of running through. Then **single-step** through the function one line at a time. After each step read the values of the variables and of any expression you have asked the IDE to **watch**, and compare them with the values you expected; the first line after which a value is wrong is where the logic error is. The **report window** shows any run-time error message and the output produced so far.

> **Worked example.** Put each feature in its syllabus group: prettyprint, context-sensitive prompt, dynamic syntax check, breakpoint, expand/collapse code blocks, report window.

Coding: context-sensitive prompt. Initial error detection: dynamic syntax check. Presentation: prettyprint, expand/collapse code blocks. Debugging: breakpoint, report window.

■ Software licences used when translating or distributing code

A software licence specifies the permitted use, copying, modification and redistribution of a copyright work. **Free Software** concerns the freedoms to run, study, change and share software; it can be sold. **Open-source** licences permit access to source and licensed modification/redistribution. Freeware can cost nothing while withholding source and modification rights. For a redistribution question, identify the actual licence's conditions; see topic 7 for comparisons and justified choices.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
operating system	software that manages the computer's hardware and resources and provides an interface between the user, the application programs and the hardware
utility software	system software that performs a specific task to maintain, optimise or protect the computer, such as a virus checker or a defragmenter
program library	a collection of pre-written, tested routines (subroutines, classes, modules) that a program can use instead of writing its own
Dynamic Link Library (DLL)	a program library whose routines are loaded into memory only when the program calls them, at run time, and are shared between programs
assembler	a translator that converts an assembly language program into machine code, one instruction for each instruction
compiler	a translator that converts the whole of a high-level language program into machine code before it is run, producing an executable file
interpreter	a translator that translates and executes a high-level language program one statement at a time

Term	Definition
integrated development environment	a single application that provides the tools for writing, translating, running and debugging a program
context-sensitive prompt	a pop-up that suggests identifiers, keywords or parameters that fit at the current point in the code
dynamic syntax check	checking the syntax of the code as it is typed and flagging an error before the program is translated
prettyprint	displaying code with keywords, identifiers and comments in different colours or fonts and with consistent indentation
breakpoint	a marked line at which the running program pauses so that variables can be inspected
single stepping	running a paused program one statement at a time under the programmer's control

Exam tips

- List the OS's jobs by their syllabus names (memory, process, hardware, file and security management) and say what each *does* – "manages resources" alone is too vague.
- Compare **compiler vs interpreter vs assembler**: what each translates, when it translates it, and how errors are reported.
- Explain what an **IDE** provides using the syllabus groups: coding, initial error detection, presentation, debugging.
- A "benefit to the developer" answer names the developer's saving: time, cost, expertise, reliability or maintenance. A "drawback" names a dependence: availability, version, fit, security.
- For "describe the operation of" a translator, give three things: what is translated (whole program or one statement), when (before running or while running), and how errors are reported (all at once or at the first error).

Common mistakes

- Writing "the OS controls the computer" or "manages resources" with no example task. Each mark is one named task with what it does.
- Saying an interpreter "compiles line by line". An interpreter translates **and executes** each statement; it never produces an executable.

- Saying a compiler runs the program. It only translates; the executable runs later, without the compiler.
- Putting a DLL "inside" the executable. That is a static library; a DLL stays a separate file loaded at run time.
- Saying defragmentation "deletes" or "compresses" files, or is needed on an SSD. It only moves blocks so each file is stored contiguously.
- Filing prettyprint or collapsing blocks under "debugging". They are presentation features; debugging is breakpoints, single stepping, watching variables and the report window.

Security, privacy and data integrity

A-Level Computer Science

Security, privacy and integrity – three different ideas

These sound alike but mean different things:

- **security** – protecting data from **unauthorised** access, change or destruction.
- **privacy** – an individual's right to **control who sees their personal data**, with consent and a clear purpose.
- **integrity** – the data being **accurate and complete** – not corrupted or accidentally changed.

A file can be secure (only the right people can open it) but lack integrity (a typo corrupted it); or accurate but not private (anyone can read it). All three are needed.

The differences the scheme wants, one sentence each: **security** is keeping the data safe from loss and from unauthorised access; **privacy** is keeping the data confidential, so that only those with the right to see it can; **integrity** is the data being correct, consistent and complete. So "the difference between security and privacy": security is about protecting the data from being accessed, changed or lost by people who should not; privacy is about the individual's right to decide who may see their personal data. "The difference between security and integrity": security protects the data from unauthorised access; integrity is about the data being accurate and up to date, which validation and verification protect.

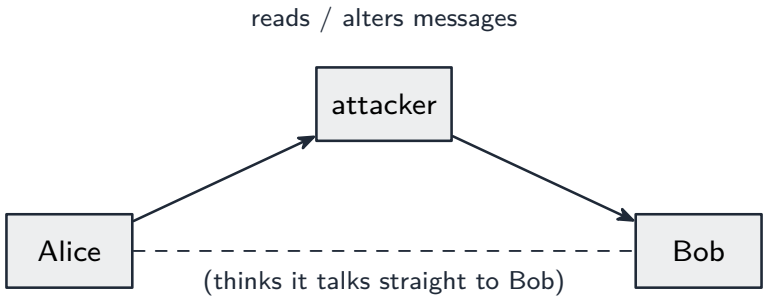
Why security matters

Two things to protect: the **data** itself (keep it confidential, intact and available) and the **computer system** (a compromised system can attack others, steal credentials, or be held to ransom).

"Why does the school need to keep both secure?" Data: it is personal and confidential, so it must not be read, changed or deleted by an unauthorised person, and its loss would stop the school working. System: an intruder who reaches the computer system can install malware, use it to attack other systems, damage the hardware or software, or lock it with ransomware; a secure system is the first line of defence for the data on it.

Threats from networks and the internet

Threats fall into three groups.



A man-in-the-middle attacker sits between the two parties

1. **Malware** (malicious software) – harmful programs:

- **virus** – self-copying code that attaches to other programs and spreads when they run.
- **worm** – self-copying code that spreads over **networks** with no user action.
- **Trojan horse** – looks useful but hides malicious code.
- **spyware** – secretly collects information (keystrokes, passwords).
- **ransomware** – encrypts your files and demands payment.
- **adware** – pushes unwanted adverts.

2. **Tricking people** (social attacks):

- **phishing** – fake emails/sites that trick users into giving credentials.
- **pharming** – redirects a user to a fake site even when they type the correct address.
- **social engineering** – tricking people into giving up information.

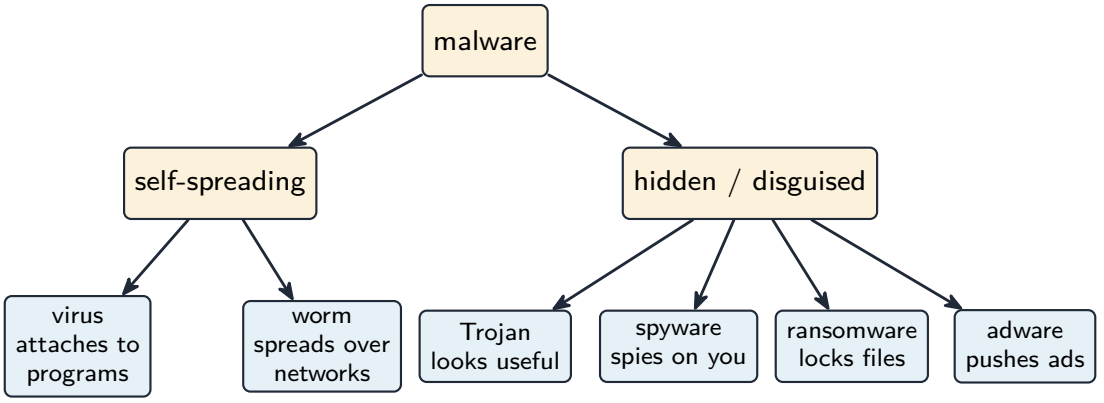
The scheme's descriptions of the four named threats: a **virus** is malicious software that replicates (copies itself), attaches itself to other files and deletes or corrupts data; **spyware** is malicious software that records the user's key presses and actions and sends them to a third party, to obtain passwords and personal data; a **phishing** email pretends to come from a legitimate organisation and contains a link to a fake website where the user is asked for personal or bank details; **pharming** is malicious code installed on the user's computer or on a web server that redirects the user to a fake website even though they typed the correct address. Similarities of spyware and a virus: both are malware, both are installed without the user's knowledge, both can send data to a third party or damage the system; the difference is that a virus replicates itself while spyware records and transmits information. Phishing and pharming both lead the user to a fake website that collects their data; phishing needs the user to click a link in an email, pharming works through code on the computer or the DNS server and needs no email.

3. **Attacks on the network:**

- **hacking** by **hackers** – unauthorised access, often via weak passwords or software flaws.
- **denial of service** (DoS/DDoS) – floods a server so real users cannot reach it.
- **eavesdropping** – capturing data in transit (a risk on open Wi-Fi).
- **man-in-the-middle** – an attacker secretly relays or alters messages between two parties.

> **Worked example.** Identify and describe two threats to the data on a school network, and give a different prevention method for each.

Threat 1, malware: a virus copied onto a computer from an email attachment or a download replicates itself and corrupts or deletes files; prevention: anti-virus software that scans files and is kept up to date. Threat 2, hacking: an unauthorised person gains access to the network, for example by guessing a weak password, and reads or changes the data; prevention: a firewall that blocks unauthorised connections, or strong passwords with two-factor authentication. A third pair, phishing: an email leads a user to a fake site that collects their login; prevention: training users to check the sender and the URL, and filtering email. The measure must match the threat: encryption does not stop a virus, and anti-virus software does not stop phishing.



Malware by behaviour: self-spreading (virus, worm) versus hidden/disguised (Trojan, spyware, ransomware, adware)

Security measures

Measures protect both the **security of data** (against loss, theft or corruption) and the **security of the computer system** (its hardware, software and network).

A standalone PC

- a strong password; **antivirus** kept up to date; prompt **software updates**; **backup** to separate media; full-disk **encryption** ; a locked screen.

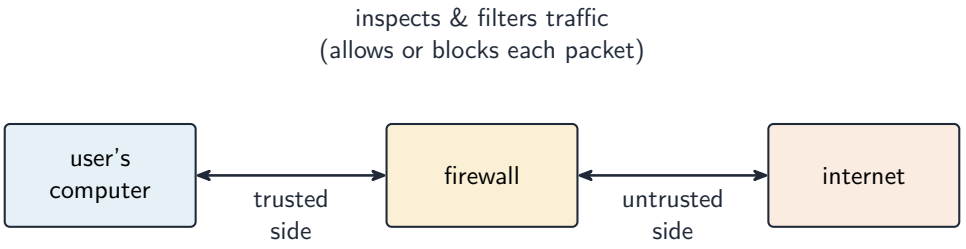
A networked PC

All the above, plus a **firewall** , per-user **permissions** (admin rights only for admins), central management of **user accounts** , and **audit logs** (who logged in, what they touched).

How the measures work, in the wording the scheme awards:

- **firewall**: examines every incoming and outgoing transmission and compares it with set criteria (a whitelist or blacklist of addresses, ports and protocols); blocks any that do not meet the criteria; can prevent access to certain sites and warn of unauthorised access attempts.
- **encryption**: the data is scrambled (encoded) with a key into ciphertext, so an intercepted copy cannot be understood without the key; the receiver uses a key to decrypt it. It protects data in transmission and in storage, but it does not stop the data being intercepted or deleted.
- **passwords and user accounts**: only a user who knows the password can log in; a strong password (long, mixed characters, changed regularly) cannot be guessed; accounts lock after repeated failures; each account carries its own access rights.
- **anti-virus and anti-spyware software**: scans files and programs against a database of known malware signatures, checks behaviour, quarantines or deletes what it finds, and must be updated so that new malware is recognised.
- **access rights**: each user (or group) is given permissions for each file or table, such as read-only or read and write, so a user cannot see or change data that is not theirs; a database can also present each user with a view containing only the fields they need.
- **biometrics**: the device captures an image of the face, fingerprint or iris, converts it to digital data, compares it with the stored data for that user and allows access only on a match; it cannot be forgotten, lent or guessed like a password.
- **backups**: a copy of the data on separate media, kept off-site, so that lost or corrupted data can be restored.

To restrict the risks of malware, in three marks: install anti-malware software and keep it updated; use a firewall; do not open attachments or download files from unknown sources; keep the operating system and applications patched; and train users.

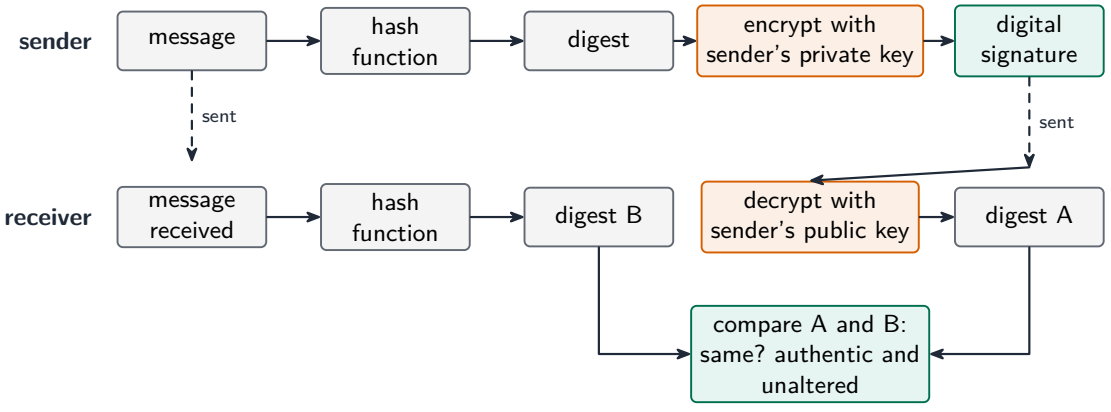


A firewall sits between the user's computer and the internet

Across the internet

- **VPN** – encrypts traffic between the user and the corporate gateway.
- **HTTPS / TLS** – encrypt web traffic.
- **digital signatures** – prove who sent a message and that it was not altered in transit.
- intrusion detection – watches traffic for known attack patterns.

How a digital signature authenticates a document (five marks): the sender puts the message through a hash function to produce a digest; the sender encrypts the digest with their **private key**, and that encrypted digest is the digital signature; the message and the signature are sent together; the receiver decrypts the signature with the sender's **public key** to recover the digest; the receiver hashes the received message and compares the two digests; if they match, the message came from the sender (only they hold the private key) and was not altered in transmission. A signature proves who sent the message and that it is intact; it does not hide the contents, which is what encryption of the message is for.



A digital signature: a hash of the message, encrypted with the sender's private key, checked by the receiver against a fresh hash

Matching measures to threats

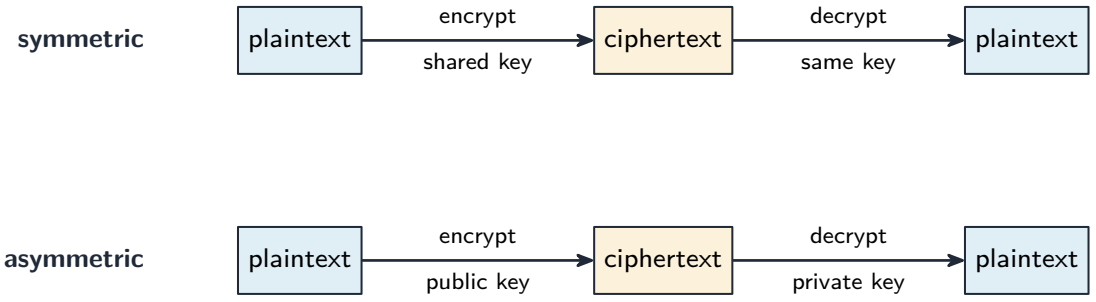
- **interception in transit** → **encrypt** the data (HTTPS, VPN). Intercepted ciphertext is useless without the key.
- **unauthorised access** → strong **authentication** (long passwords; **two-factor authentication** with a phone code or key); user **authorisation**; lock-out after failed logins.
- **malware** → **anti-virus software** and **anti-spyware** with real-time scanning; patching; avoid untrusted downloads.
- **phishing** → user training; email filtering; check the URL before entering credentials.
- **internal threats** → the **least-privilege** principle (give each user only what they need); auditing.
- **DDoS** → rate limiting and traffic filtering.

For confidential data crossing the internet, the scheme's method is **encryption**: the data is encoded with a key into ciphertext, so that an unauthorised person who intercepts it cannot read it, and only the intended receiver, who has the key, can decode it. For a program file sent by email for testing, the same answer applies (encrypt the file, or send it over an encrypted connection), together with a password on the file itself.

Protecting the data itself

- **encryption** – turn **plaintext** into **ciphertext** with a key. **Symmetric encryption** (AES) uses one shared key; **asymmetric encryption** (RSA) uses a **public key** and a **private key**. Protects data at rest and in transit.
- **access control** – file permissions (read/write/execute) and **access rights**, enforced by the OS.
- **authentication** – **authentication techniques** verify the user: something you know (password), have (token, phone), or are (**biometrics** – fingerprint, face, iris); strongest combined.
- **backups** – keep copies (some **off-site**) so loss or corruption is recoverable.
- **physical security** – locked server rooms, cable locks.

Access rights in a database, described for three marks: each user is given an account with a username and password; the database administrator assigns each account permissions for each table, such as read-only, read and write, or no access; users see only the tables and fields they are allowed to, so a customer cannot open the staff table and a clerk can read but not change the prices. The **DBMS** enforces this with its access rights and with **views**, and it can encrypt the stored data as well.



Symmetric uses one shared key; asymmetric uses a public key to encrypt and a private key to decrypt



A security token shows a changing code for two-factor authentication ("something you have")



A fingerprint reader checks "something you are" – a feature of the person, not a password

Data integrity

Data has integrity when it is accurate and complete. Two techniques: **data validation** (catch bad data before storing) and **data verification** (confirm data was entered or transferred correctly).

■ Validation – does the data make sense?

Validation checks data against sensible rules, automatically:

- **range check** – within limits (a month is 1–12).
- **limit check** – on the correct side of a single limit (e.g. age ≥ 18).
- **existence check** – the referenced item exists (e.g. a product code is in the table).
- **length check** – the right number of characters.
- **type / character check** – the right kind of data (a phone field allows only digits).
- **format check** – matches a pattern (an email must contain `@`).
- **presence check** – required fields are not empty.
- **check digit** – an extra digit computed from the others (ISBN, card numbers) that spots transcription errors.

> **Worked example.** In a simple check-digit scheme the check digit is the remainder when the sum of the digits is divided by 10, appended to the number. The number 4162 has digit sum 13, so it is stored as 41623. A user types 14623: the first two digits are swapped, but the sum is still 13, so the check digit still matches and the error is **not** caught. A user who types 41523 is caught, because $4+1+5+2 = 12$ gives check digit 2. Weighted schemes can detect some transpositions that this unweighted rule misses. Detection depends on the weights, the positions and the modulus: an ISBN-13 rule, for example, does not detect every possible transposition. A check digit is validation: it tests the number against a rule at the moment it is entered.

- **lookup check** and **consistency check** (e.g. delivery date $\geq$ order date).

Validation catches data that is wrongly **formatted**, but not data that is the right format yet factually wrong ("Bob" for "Bib").

> **Worked example.** Identify the validation check each piece of pseudocode performs.

Pseudocode	Check
`IF x < 0 OR x > 10 THEN OUTPUT "Invalid"`	range check: the value must lie between two limits
`IF x = "" THEN OUTPUT "Invalid"`	presence check: the field must not be empty
`IF NOT(x = "Red" OR x = "Yellow" OR x = "Blue") THEN OUTPUT "Invalid"`	lookup (existence) check: the value must be one of a list
`IF LENGTH(x) <> 6 THEN OUTPUT "Invalid"`	length check: the right number of characters
`IF MID(x, 1, 1) < "A" OR MID(x, 1, 1) > "Z" THEN OUTPUT "Invalid"`	format check: a particular character must be a letter

To validate a car registration number that must be one letter, three digits and two letters: a **format check** tests each position against its pattern, and a **length check**

confirms six characters. To validate a date of birth: a format check (`DD/MM/YYYY`), a range check (the month is 1 to 12, the year is not in the future) and a presence check (it is not left blank). A mark between 0 and the maximum for the test needs a type check (an integer) and a range check, with the upper limit read from the test's own record: that is how validation protects integrity, by refusing data that could not be correct.

■ Verification – was the data entered or transferred correctly?

Verification checks the data was not changed in moving from one place to another.

During entry: **double entry** (type it twice and compare, as for a new password) or **visual** check.

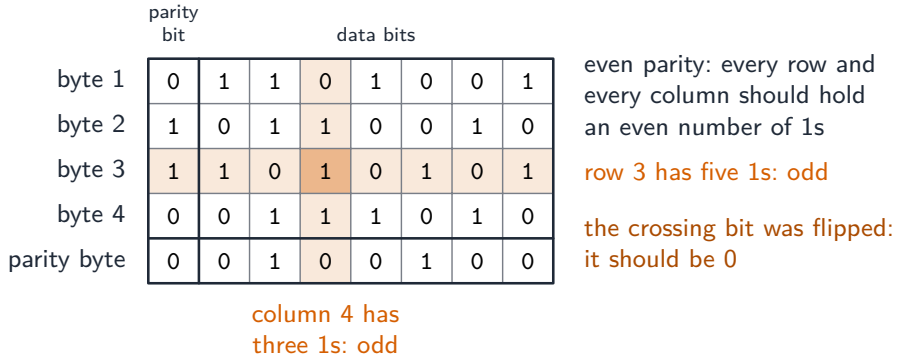
In the scheme's words, **double entry** is entering the data twice, by the same person or by two people, and having the computer compare the two versions and report any difference; a **visual check** is the person comparing what is on the screen with the original source document and correcting any difference before saving. Both protect integrity by making sure the stored data matches the source. Even after validation and verification the data can still be wrong: it can be sensible and match the source, yet the source itself was wrong, or the user typed a different but valid value from the one intended.

During transfer (bits can flip):

- **parity check** – an extra bit makes the number of 1s even (even parity) or odd. The receiver re-counts. Catches single-bit errors.
- **checksum** – the sender sends a summary value of the data; the receiver recomputes it and compares.
- **cyclic redundancy check** (CRC) – a stronger checksum using polynomial division, catching many more error types.

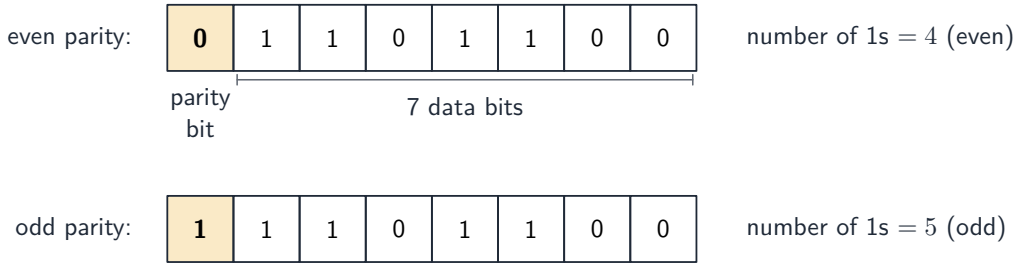
A **parity block check** goes further and *locates* the error. Arrange the bytes in a grid: give each byte a **row** parity bit, then compute one extra **parity byte** whose bits are the **column** parity of the bytes above. A single flipped bit now fails **one row and one column** – their intersection pinpoints exactly which bit changed, so it can even be corrected.

> **Worked example.** Four bytes are sent with even parity, followed by a parity byte. Find the bit that was corrupted.

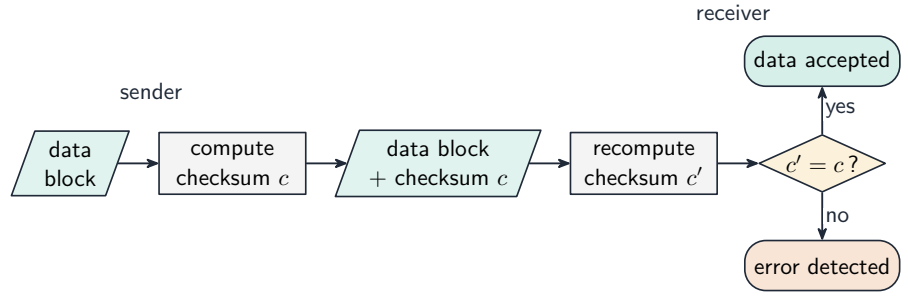


A parity block check: the row that fails and the column that fails cross at the flipped bit

Count the 1s in each row and each column. Every row and column should have an even number; byte 3 has five and column 4 has three. The bit where that row and that column cross is the one that changed, so it is reset from 1 to 0. A parity check on its own detects an error in a byte but cannot say which bit; two errors in the same byte cancel and pass unnoticed. A **checksum**, explained for three marks: the sender puts the block of data through an algorithm that produces a checksum value; the data and the checksum are sent together; the receiver runs the same algorithm on the data it received; if the two checksums match, the data is accepted, and if not, it is rejected and sent again.



The parity bit is set to make the number of 1s even or odd



worked example – checksum = (sum of bytes) mod 256

1 sum of all bytes: $X = 1185$

2 $X \div 256 = 4.629$, round down: $Y = 4$

3 $Z = Y \times 256 = 1024$

4 checksum $= X - Z = 1185 - 1024 = 161$

Working out a checksum for a block of data

Verification only proves what arrived matches what was sent – not that the data is correct, and not against deliberate tampering. **Validation** asks "is this sensible?"; **verification** asks "was this copied correctly?" – use both.

The table questions sort the methods by when they are used: **during data entry**, double entry and a visual check; **during data transfer**, a parity check (byte or block) and a checksum. Transferring video files from a camera to a server uses a checksum: the camera computes it, the server recomputes it, a mismatch means retransmit.

validation **"is this data sensible?"**	**verification** **"was it copied correctly?"**
checks rules *before* storing: · range / type / format · length / presence · check digit	checks the data did not change: · double entry · parity check · checksum / CRC
catches nonsense data	**catches copying errors**

Validation checks the data makes sense; verification checks it was copied without change

Worked example. A user types their date of birth as `31/02/2009`, and types their email address twice. Which check catches which error, and what is the difference? **Validation** asks "is this data **sensible**?" - the computer tests it against a rule, and a format or range check rejects `31/02/2009` because February never has 31 days. **Verification** asks "was this data **entered correctly**?" - typing the email twice is **double entry**, and comparing the two copies catches a typing slip. The limit is what makes this a favourite question: validation can never tell you the data is **right**, only that it is **possible** - `01/02/2009` passes every validation rule even if the user was actually born on a different day. Say what each check **can** and **cannot** catch.

A proxy server in a security answer

A **proxy server** handles requests on a client's behalf. An organisation may configure it to filter access to websites, log requests or cache responses. Explain the configured protection: merely naming a proxy does not show how it prevents the stated threat. A router forwards packets between networks; these roles are different.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly.

Term	Definition
data security	keeping data safe from loss and from unauthorised access, change or deletion
data privacy	keeping data confidential, so that it is seen only by those who have the right to see it
data integrity	the data being accurate, consistent and complete
malware	malicious software that is installed without the user's knowledge to damage a system or steal data
virus	malware that replicates itself, attaches to other files and corrupts or deletes data
spyware	malware that records the user's key presses or actions and sends them to a third party
phishing	an email pretending to be from a legitimate organisation that leads the user to a fake website to collect personal data
pharming	malicious code that redirects the user to a fake website even when the correct address is entered
firewall	hardware or software that examines all traffic entering or leaving a system against set criteria and blocks what does not meet them

Term	Definition
encryption	scrambling data with a key into ciphertext, so that it cannot be understood without the key to decrypt it
digital signature	a hash of a message encrypted with the sender's private key, used to prove who sent it and that it was not altered
data validation	an automatic check that entered data is reasonable and follows set rules
data verification	a check that data has been entered or transferred correctly, by comparing it with the source or with a recomputed value
check digit	an extra digit calculated from the other digits of a number and appended to it, so that an error in the number can be detected
parity check	an extra bit added to a byte so that the number of 1s is even (or odd), which the receiver recounts
checksum	a value calculated from a block of data by an algorithm and sent with it, recalculated by the receiver and compared

Exam tips

- Keep the three ideas separate: **security** (keeping data safe), **privacy** (who may see it), **integrity** (keeping it correct).
- Match each **threat** (malware, hacking, phishing, interception) to a **measure** (firewall, encryption, authentication, access rights).
- Encryption protects **confidentiality, not integrity** – use a checksum, parity or check digit for integrity.
- Distinguish a **virus, worm and Trojan** and how each spreads.

Common mistakes

- Giving the same measure for two threats, or a measure that does not fit the threat. Each threat in the table needs a different prevention that actually stops it.
- Naming a measure without saying how it works. "Firewall" scores when it is followed by "compares traffic with set criteria and blocks what fails".
- Calling validation a check that the data is correct. Validation checks that data is reasonable; verification checks that it matches the source. Neither proves it is true.
- Saying a digital signature encrypts the message. It encrypts a hash of the message with the private key; the receiver decrypts it with the public key and compares hashes.
- Describing a check digit as verification, or a parity check as validation. The check digit is a validation rule on entry; parity and checksums verify a transfer.

- Writing that a virus “sends data to a third party” and spyware “replicates”. The replicating one is the virus; the recording one is spyware.

Ethics and Ownership

A-Level Computer Science

Ethics for computing professionals

A **computing professional** is someone whose work – software, systems, networks, data – affects other people. Because the work is technical, others often cannot judge whether it was done well or honestly. So the profession follows shared **ethics** (principles for good behaviour).

Why ethics matters

- **trust** – users and employers trust professionals to act in their interest. Without that trust, software loses credibility.
- **impact** – software runs medical devices, banking, vehicles. Careless or dishonest work can hurt people.

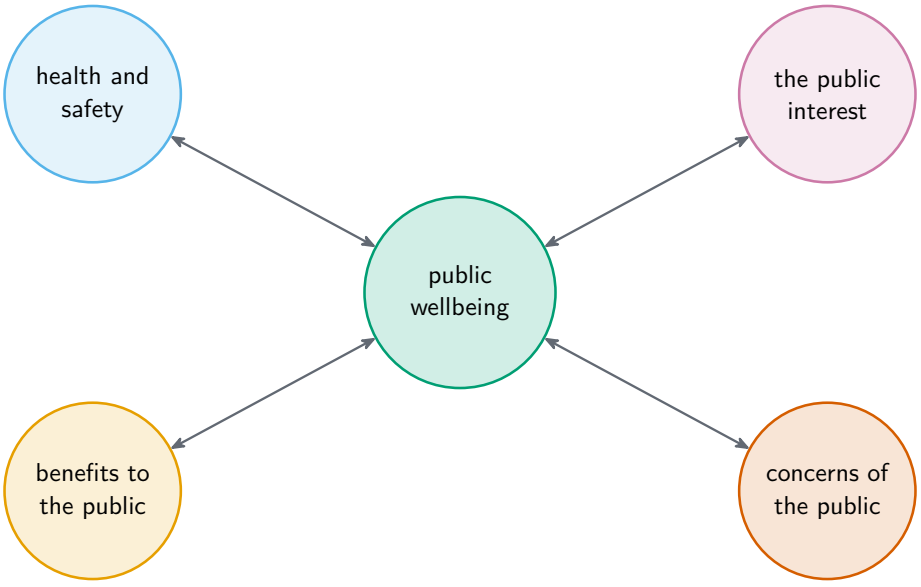
Professional bodies (BCS, ACM, IEEE) publish codes of ethics for members.



CCTV raises privacy concerns – one of the ethical issues a computing professional must weigh



Discarded electronics (e-waste) are a growing environmental cost of computing



Software development affects the public's wellbeing in several ways

Typical principles

- **public interest first** – protect the safety and welfare of those affected.
- **honesty and competence** – be honest about your skills; don't claim expertise you lack.
- **confidentiality** – protect clients' and employers' private information.
- avoid **conflicts of interest** – don't take work where your interest clashes with the client's.
- keep your skills current; respect **intellectual property** and **privacy**; treat colleagues fairly.

Joining a professional body

The syllabus names two: the **BCS** (British Computer Society) and the **IEEE** (Institute of Electrical and Electronics Engineers). Both publish a **code of conduct** that members agree to follow. The benefits of joining, in the scheme's words: a set of ethical guidelines to follow, so decisions are not left to personal judgement; training, conferences and publications that keep the member up to date; advice and support, including legal help, when a problem arises; and recognised professional status, so employers and clients trust the member's work. The consequences of **not** joining: no guidance on ethical decisions, so the programmer may act unethically without realising; less credibility with employers and customers, so it is harder to win work; no support in a dispute; and being out of date with developments and law. The purpose of a code of conduct (two marks): to create a safe, respectful and professional working environment, and to make sure every employee understands what is expected and the

consequences of their actions.

> **Worked example.** Explain why a programmer needs to act ethically towards colleagues and towards the public.

Colleagues: treat them fairly and without discrimination; respect their work, their ideas and their confidential information; be honest about mistakes and give credit where it is due; support their development rather than undermine it. The public: protect their personal data and privacy; produce software that is safe, reliable and properly tested, because faults can cause harm; be honest about what the software can do; take on only work within your competence; obey the law and consider the wider effects on society and the environment. Each side earns marks for a reason and its consequence, not for the word "fair" alone.

■ Acting ethically vs unethically

Acting **ethically** protects users, strengthens reputation, reduces legal risk, and builds trust. Acting **unethically** (skipping testing, hiding bugs, misusing data) can harm real users, lead to dismissal or legal action, damage reputation, and erode trust in technology generally.

When you face a borderline decision: identify whose interests are affected, check the code of ethics and the law, weigh the consequences, ask a trusted senior, and choose the option that protects users above short-term convenience.

> **Worked example.** Your team's new AI hiring tool sorts CVs ten times faster, but you notice it rejects more older applicants. Shipping it pleases your manager, but it treats one group unfairly. The ethical choice is to **hold it back until the bias is fixed** – public interest and fairness come before short-term convenience.

Ethics also applies to users. A student who connects a personal computer to the school network should respect other people's privacy and data, not use social media inappropriately or bully others, not download or share copyrighted material, not introduce malware or try to access systems they are not allowed to, and use the network for the purpose it was provided. A "give three ethical considerations" answer lists three of these.

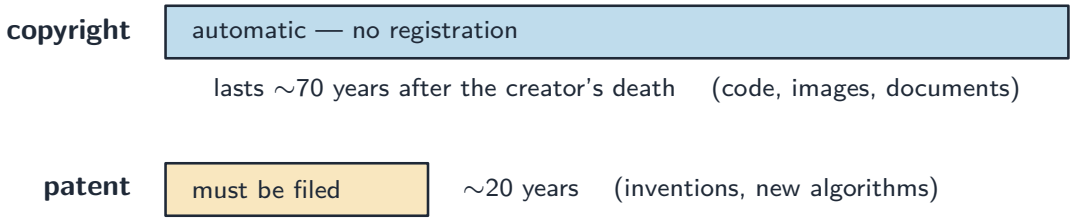
Copyright

Copyright is the legal right of the creator of an original work to control how it is **copied, distributed, modified and performed**. It applies automatically (no registration) to source code, software, documents, images, audio and video.

Without copyright, anyone could copy software freely, the developer would not be paid, and plagiarism would be legal. With copyright, developers can earn from their work (encouraging more software), users know who made it, and re-use happens on the developer's terms through licensing. Copyright lasts a long time (often 70 years after the creator's death). General ideas and algorithms are not covered by copyright

but may be covered by a **patent** .

Why a programmer should copyright a program, in the scheme's words: to be identified as the owner and author (formal recognition of ownership); so that there are legal consequences if anyone copies or steals it; to restrict competitors from selling the same work; and to be able to earn money by licensing it. Copyright applies to the program as written; a different program that does the same job does not infringe it.



Copyright is automatic and long-lasting; a patent must be filed and lasts about 20 years

Software licences

A **software licence** is a contract granting permission to use software on the owner's terms; choosing and applying one is called **software licencing**.

Commercial (proprietary)

- **commercial software** is sold: you **buy a licence**; the software is used only within its terms.
- the source code is not given (a **proprietary** product); you cannot modify or redistribute it.
- examples: Microsoft Office, Adobe Photoshop, most games.

Used when the developer wants **revenue** per user and to **keep control** of the code.

Open-source

- the source code is **public**; users can read, modify and redistribute it (**open-source**).
- **permissive** licences (MIT, BSD) allow almost any use; **copyleft** licences (GPL) require that modified versions are released under the same licence ("share-alike").
- the **Free Software Foundation** (FSF) and the **Open Source Initiative** (OSI) promote and approve open-source licences.

The syllabus names both, and they are marked as distinct answers. **Free Software** (the FSF's term) means free as in freedom, not price: the user may run the program for any purpose, study and change it (so the source code must be available), redistribute

copies, and distribute modified versions; a fee may still be charged for a copy. **Open Source** (the OSI's definition) requires that the source code is available, that the program may be modified and redistributed, and that the licence does not discriminate against any person or field of use. "Identify two licence types that let other people edit and redistribute the program" is answered with these two.

- examples: Linux, Python, Apache.

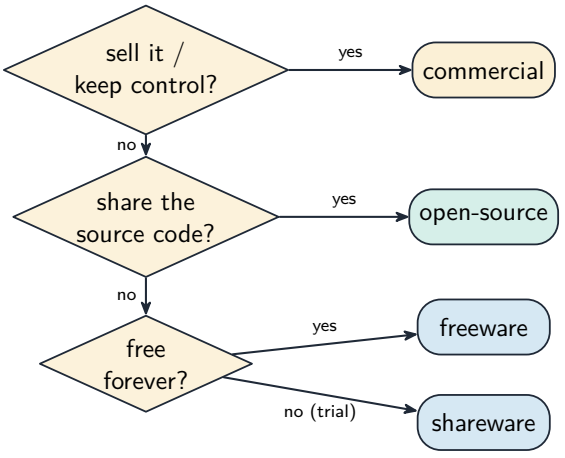
Used when the developer wants the software **widely used and improved by the community**.

■ Freeware and shareware

- **freeware** – free of charge, no source code, may be redistributed but not modified (Acrobat Reader, WhatsApp).
- **shareware** – free for a trial period, then you pay to keep using it; no source code.

The scheme's descriptions: **shareware** is distributed free for a trial (a limited time or limited features) and the user pays to continue using the full version; **commercial software** is sold for a fee, the source code is not supplied, the licence protects the developer's intellectual property, and the fee usually buys support and updates. Benefits of shareware to the programmer: users can try the program before buying, so they are more likely to purchase; it spreads widely at almost no advertising cost; and those who keep it pay. Benefits of a commercial licence: the developer earns a fee for every copy; the code and its rights stay protected; and the income funds support, updates and further development.

Type	Cost	Source	Redistribute	Modify
Commercial	Paid	No	No	No
Open-source	Free	Yes	Yes	Often, with conditions
Freeware	Free	No	Yes	No
Shareware	Free trial, then paid	No	Sometimes	No



Choosing a licence from the developer's goal

To justify a licence choice, link it to the developer's goal (revenue, reach, community), the user's needs (cost, customising), and the use case.

> **Worked example.** A programmer has written a game to sell to the public. Identify the most appropriate licence and justify it.

A commercial licence: the game is sold for a fee, so the programmer earns money from every copy; the source code is not released, so nobody can copy the game or change it and sell it as their own; the licence protects the intellectual property; and buyers receive updates and support. Open source would not fit, because the source code would be available, so the game could be copied, changed and redistributed without payment.

> **Worked example.** A program helps shoppers by reading product labels aloud. Explain why an open source licence might not be appropriate.

The source code would be accessible, so it could be changed; a changed version might output the wrong product information, so shoppers could buy the wrong item; and the programmer would lose control over the quality and safety of what is distributed under the program's name. Going the other way, programs are released as open source so that other developers can improve and extend them, so that they are adopted widely at no cost, and so that users can adapt them to their own needs.

Artificial Intelligence (AI)

Artificial intelligence builds systems that do tasks once thought to need **human intelligence** – recognising speech and images, translating, playing games, driving.

Most modern AI uses **machine learning** – algorithms that improve at a task by learning patterns from large amounts of data, instead of being programmed

step by step. **Deep learning** , using **neural networks** with many layers, is the leading approach today.

Everyday examples

AI tasks split into two kinds – **understanding input**, and **producing output or decisions**.

Understanding input:

- **speech recognition** – spoken words to text (voice assistants).
- **image recognition** – finding objects, faces or text in images.

Producing output or decisions:

- **machine translation** – automatic translation between languages.
- **recommendation systems** – suggesting products, videos or music.
- **autonomous vehicles** and robots.

A common exam scenario: a program reads a label with a camera, translates it, and reads it aloud – using **optical character recognition** to find the words, machine translation to convert them, and **text-to-speech** for the audio.



reading a foreign label aloud

A common scenario: OCR → machine translation → text-to-speech reads a foreign label aloud

A four-mark "explain how AI is used" answer follows the pipeline step by step: image recognition (OCR) analyses the pixels of the photograph to locate the characters; the patterns of pixels are converted into individual characters and words; machine translation converts the words into the user's language; and text-to-speech produces the spoken output. Each step is a mark.

Benefits

- **accessibility** – speech/image AI helps users with impairments; translation helps non-native speakers.
- **productivity** – automating repetitive tasks frees people for creative work.
- **decision support** – AI spots patterns in huge datasets (medical diagnosis, fraud detection).
- **always available**, and **personalised** to each user.

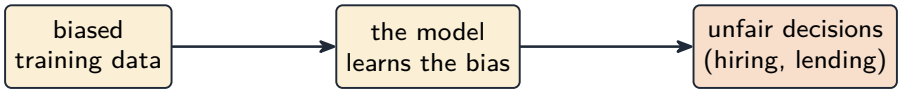
Impacts: social, economic, environmental

The syllabus asks for the impact of AI under three headings, and a question names one of them. Give the impact **and** its consequence.

- **Social**: benefits – a label-reading program helps people with a visual impairment, people who cannot read the language, and people with reading difficulties; facial recognition at an airport speeds up identity checks and can stop wanted people entering. Harms – facial recognition can misidentify people and tracks everyone without consent, so privacy is lost; students who use AI to do their homework may not develop reasoning and problem-solving skills, may rely on it instead of learning, and may lose the collaboration and face-to-face communication that working together brings.
- **Economic**: an AI fault-diagnosis module in a repair garage diagnoses faults faster and more accurately, so more vehicles are repaired per day and costs fall; but fewer skilled mechanics may be needed, so jobs are lost, and the module must be bought and maintained. More generally, AI raises productivity and creates new jobs in some fields while removing routine jobs in others.
- **Environmental**: training and running large models uses a great deal of electricity and water for cooling in data centres, and the hardware becomes e-waste; on the other side, AI is used to cut energy use in buildings, optimise transport and monitor the environment.
- **Ethical** (the classroom question): an AI that marks work or watches students must be fair to every student, must not leak their data, must be explainable when it makes a decision about them, and must not replace the judgement of a teacher where that matters.

Concerns

- **bias** – unfair patterns in the training data become unfair AI decisions (hiring, lending).
- **job displacement** – AI may replace some roles.
- **privacy** – training often uses large amounts of personal data.
- **transparency** – large models are "black boxes", hard to explain.
- **accountability** – when AI is wrong, who is responsible: developer, user, or operator?
- **misuse** – deepfakes, misinformation, surveillance.



How bias gets into AI: biased data → a biased model → unfair decisions

Professionals must **understand the limits** of the AI they build, **inform users**, and

reduce harm.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly.

Term	Definition
ethics	the moral principles that govern how a person behaves, in a profession the standards set out in its code of conduct
code of conduct	the rules an organisation or professional body sets for how its members must behave
copyright	the legal right of the creator of an original work to control how it is copied, distributed and modified
software licence	the legal agreement that states how a piece of software may be used, copied and distributed
commercial software	software sold for a fee, without source code, under a licence that protects the developer's intellectual property
free software (FSF)	software whose users are free to run, study, change and redistribute it, so its source code is available
open source (OSI)	software whose source code is available and which may be modified and redistributed under its licence
shareware	software distributed free for a trial period or with limited features, after which the user pays to continue
freeware	software that is free of charge to use and copy, but whose source code is not released and may not be modified
artificial intelligence	computer systems that perform tasks which normally need human intelligence, such as recognising images and speech
machine learning	a form of AI in which a system improves at a task by learning patterns from data rather than by explicit programming

Exam tips

- Answer ethics questions against a **professional code of conduct** (public interest, competence, honesty), not personal opinion.
- Distinguish **copyright** (protects the expression) from a **patent** (protects an invention).

- Compare software licences: proprietary, open-source, freeware, shareware and FOSS.

Common mistakes

- Giving a personal opinion ("it is wrong") instead of a reason with a consequence ("faulty software could harm users, so it must be tested").
- Treating free software and freeware as the same thing. Free software is about the freedom to study and change the code; freeware is merely free of charge.
- Saying open source means free of charge. It means the source code is available and may be modified and redistributed; a fee may still be charged.
- Writing that copyright must be registered. It applies automatically to the work as written.
- Naming an impact without its consequence. "Job losses" scores when it is tied to why: the AI does the diagnosis, so fewer mechanics are needed.
- Describing what AI is instead of how it is used. The marks are for the steps: recognise, convert, translate, speak.

Facial recognition in context

A camera captures an image; software detects a face and extracts features. An AI model compares these features with stored examples or a template and returns a match or confidence score. The system applies a threshold to decide whether to accept the match. Training uses many labelled examples; recognition uses the trained model on a new image. Poor lighting, biased training data and similar faces can cause false matches. Access control therefore needs a fallback check and protection of biometric data.

A light sensor can measure ambient light; an ADC converts its analogue output to digital data so a controller can adjust illumination against a threshold. Distinguish this sensor input from the camera image used to identify a person.

Free Software and Open Source definitions overlap: GPL and MIT are examples accepted under both. Free Software describes users' freedoms to run, study, modify and redistribute; it does not require a zero selling price. GPL is copyleft; MIT is permissive. Check the actual licence conditions rather than treating FSF and OSI as opposite licence types.

Databases

A-Level Computer Science

File-based storage and its limits

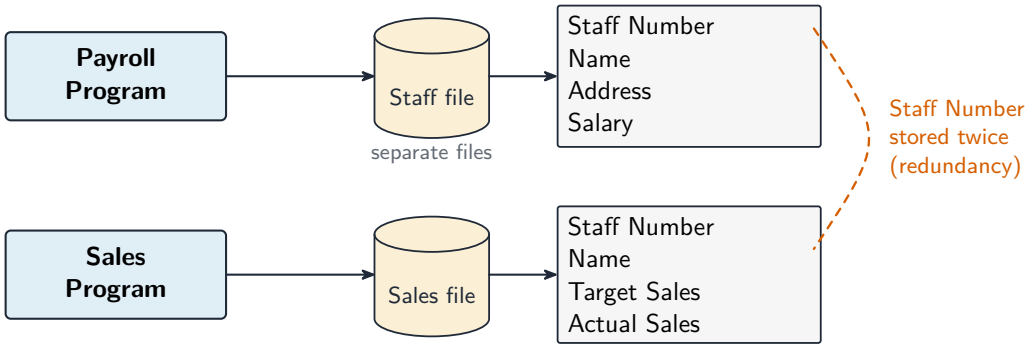
Before databases, programs stored data in **flat files** – usually one file per program. This is fine for small data but breaks down at scale.



File-based storage keeps data in separate files, like papers in a filing cabinet – hard to search and easy to duplicate

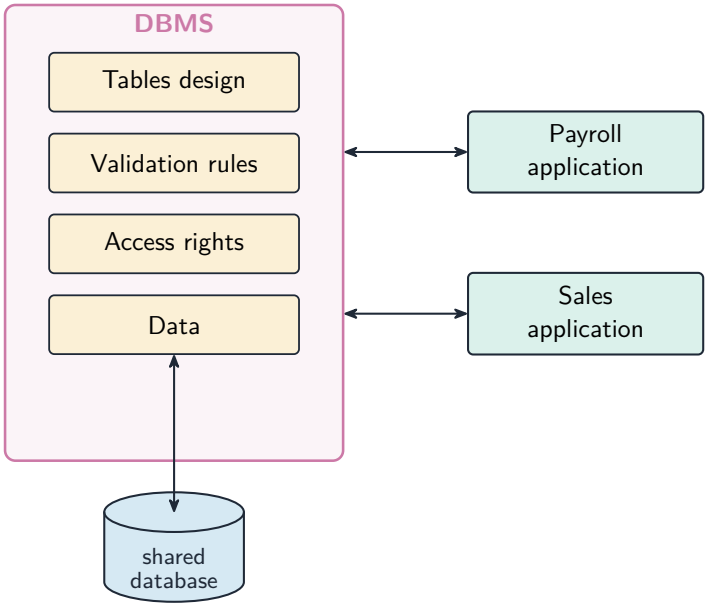
Limitations

- **data redundancy** – the same data (a customer's address) is held in several files, one per program, so storage is wasted and every copy must be updated.
- **data inconsistency** – when one copy is updated and another is not, the files disagree and nobody knows which is right.
- **data dependence** – each program is written for the exact layout of its files; change a field's length or add a field and every program that reads the file must be rewritten.
- **no shared access** – a file is locked while one program uses it, so users cannot work on the data at the same time.
- weak **integrity** – no central rules stop an invalid value or a link to a customer who does not exist; weak **security** – access is per file, not per field; and **queries** across files need a new program each time.



The file-based approach: each program keeps its own files

A **relational database** fixes these by storing data in **tables** managed by one piece of software (the DBMS) that all programs use.



The database approach: one DBMS serves all the programs

Why a relational database is better – the three-mark answer. Each item of data is stored **once**, in one table, and tables are linked by keys, so there is no redundancy and no inconsistency; the data is **independent** of the programs, which ask the DBMS for what they need and are unaffected when the structure changes; and the DBMS enforces **integrity** rules, controls **access** per user and per field, allows **many users at once**, and answers any **query** without a new program being written.

> **Worked example.** A repair shop stores its customers, devices and repair jobs using a file-based approach, one file per program. Give three problems this causes, and describe how a relational database would remove them.

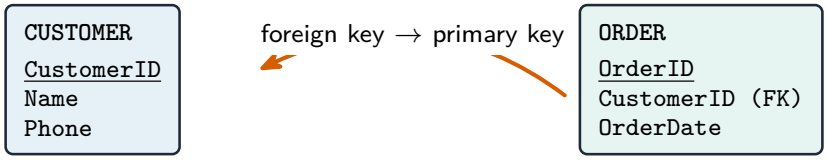
The customer's name and phone number are stored in the repairs file *and* the invoices file (redundancy); when a customer changes number, one file is updated and the other is not (inconsistency); and when the shop wants a new report – repairs per technician – a new program has to be written to read the files (no ad-hoc queries). In a relational database the customer is stored once in a CUSTOMER table and referred to by CustomerID from the REPAIR table, so a change is made once and is seen everywhere; the report is a single SQL query.

Relational model – terms

- **table** (relation) – a grid of rows and columns; one table per type of **entity** (e.g. `CUSTOMER`).
- **record** (row, also called a **tuple**) – one row; one instance of the entity.
- **field** (column, also called an **attribute**) – one column; one piece of information about each record.
- **primary key** – a field (or fields) that **uniquely identifies** each record; never null or duplicated.
- **foreign key** – a field whose value matches the primary key of another table, linking the two.
- **composite key** – a primary key made of two or more fields together.
- **candidate key** – any field(s) that could be the primary key.
- **secondary key** – a non-primary field that is indexed for fast searching.
- **indexing** – building an index on a field so look-ups and joins run faster.
- **referential integrity** – every foreign-key value must match an existing primary key (no orphan records).

A table is written in shorthand with the primary key underlined and foreign keys noted:

```
CUSTOMER(CustomerID, Name, Phone)
ORDER(OrderID, CustomerID, OrderDate)   -- CustomerID is FK → CUSTOMER
```



A foreign key links two tables: ORDER.CustomerID matches the primary key CUSTOMER.CustomerID

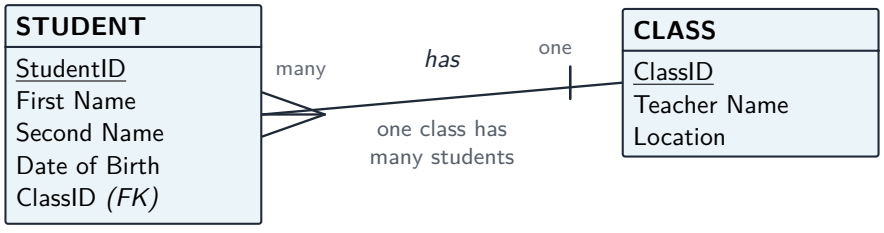
Worked example. State what is meant by *entity*, *primary key* and *referential integrity* in a relational database, and complete the term ↔ description table for *tuple* and *attribute*.

An **entity** is something about which data is stored – a person, object or event – which becomes one table. A **primary key** is the attribute (or combination of attributes) that uniquely identifies each record in a table. **Referential integrity** means that every foreign-key value must match the value of a primary key in the table it refers to, so a record cannot refer to one that does not exist. A **tuple** is one row of a table (one record); an **attribute** is one column (one field). Learn the pairs: table/relation, record/tuple, field/attribute.

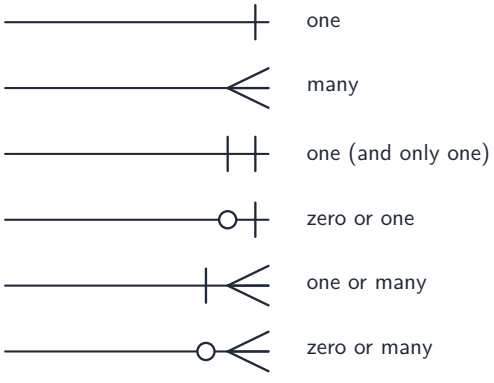
Entity-relationship (E-R) diagrams

An **entity-relationship diagram** shows the structure: each entity is a rectangle, each relationship a line, with the **cardinality** marked at each end:

- **one-to-one** (1:1).
- **one-to-many** (1:M) – each Customer has many Orders; each Order has one Customer.
- **many-to-many** (M:N) – Students take many Courses, and Courses have many Students.



An E-R diagram: one class has many students



Crow's-foot symbols for the cardinality of a relationship

A many-to-many relationship cannot be stored directly. Break it into two one-to-many relationships through a **link table** holding the two foreign keys:

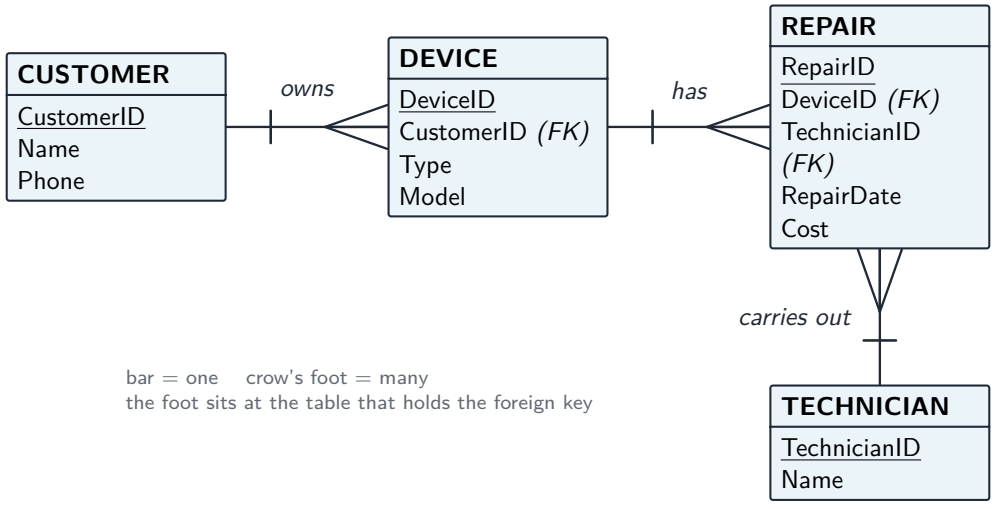
```
ENROLMENT(StudentID, CourseID, EnrolmentDate)
```



a link table turns many-to-many into two one-to-many relationships

A link table resolves a many-to-many relationship into two one-to-many relationships

Drawing the E-R diagram for a given set of tables. Each table becomes an entity. A relationship exists wherever one table holds a **foreign key** to another; it runs from the table holding the foreign key (the *many* end) to the table whose primary key it is (the *one* end). A table with two foreign keys and no other identity is usually a **link table** resolving a many-to-many relationship. Label each line with the relationship type.



Drawing the diagram from the tables: every foreign key is a one-to-many relationship, with the "many" at the table that holds it

Worked example. A repair shop has the tables `CUSTOMER(CustomerID, Name, Phone)`, `DEVICE(DeviceID, CustomerID, Type, Model)`, `TECHNICIAN(TechnicianID, Name)` and `REPAIR(RepairID, DeviceID, TechnicianID, RepairDate, Cost)`. Identify the relationships and their types.

`DEVICE` holds `CustomerID`, so CUSTOMER–DEVICE is **one-to-many** (one customer, many devices). `REPAIR` holds `DeviceID`, so DEVICE–REPAIR is **one-to-many**; it also holds `TechnicianID`, so TECHNICIAN–REPAIR is **one-to-many**. There is no direct CUSTOMER–REPAIR line: the link runs through DEVICE. Three lines, three crow's feet, all at the REPAIR or DEVICE ends.

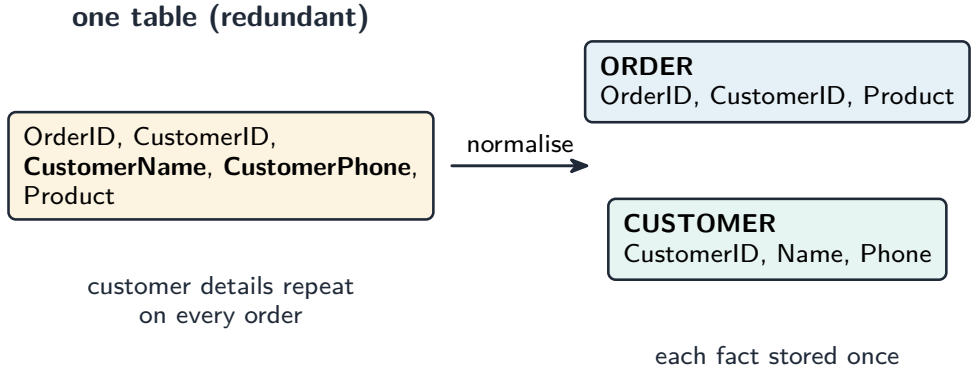
Normalisation

Normalisation organises tables to **cut redundancy and inconsistency**, going through **normal forms** in order.

- **First normal form (1NF)** – every field holds a single (**atomic**) value, with no repeating groups, and a primary key.
- **Second normal form (2NF)** – in 1NF, and every non-key field depends on the **whole** primary key (only matters for a composite key).
- **Third normal form (3NF)** – in 2NF, and every non-key field depends **only** on the primary key, not on another non-key field (no **transitive dependency**).

A 3NF design stores each fact once, so insert/update/delete anomalies disappear. The trade-off is more tables and more joins. Aim for 3NF.

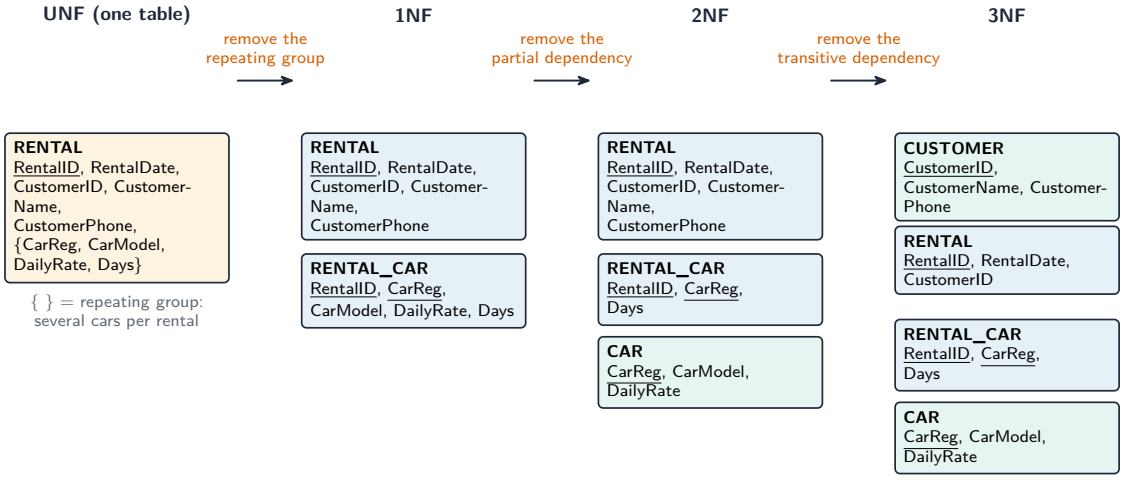
To produce a 3NF design: find the entities and their attributes; choose a primary key for each; split repeating/non-atomic fields (1NF); split fields depending on part of a composite key (2NF); split fields depending transitively on the key (3NF); add foreign keys for the relationships.



Normalisation removes redundancy by splitting repeated data into its own table

Worked example. The table `ORDER(OrderID, CustomerID, CustomerName, ProductID, Quantity)` has the composite primary key `(OrderID, ProductID)`. Normalise it to 3NF. Test each non-key field against the key. `Quantity` depends on **both** `OrderID` and `ProductID`, which is fine. But `CustomerID` depends on `OrderID` **alone** - only *part* of the composite key. That is a **partial dependency**, so the table is not in **2NF**. Split it into `ORDER_LINE(OrderID, ProductID, Quantity)` and `ORDER(OrderID, CustomerID, CustomerName)`. Now test 3NF: in that new `ORDER` table, `CustomerName` depends on `CustomerID`, which is **not** the key - a **transitive dependency**. Split again: `ORDER(OrderID, CustomerID)` and `CUSTOMER(CustomerID, CustomerName)`. Name the **dependency** that breaks each form (partial breaks 2NF, transitive breaks 3NF); "it has repeated data" describes the symptom and earns nothing.

The three questions to ask of any table. *Is every cell a single value, with no repeating group?* If not, it is not in 1NF. *If the key is composite, does every non-key field depend on the whole key?* If some field depends on part of it, there is a **partial dependency** and the table is not in 2NF. *Does every non-key field depend on the key alone?* If a field depends on another non-key field, there is a transitive dependency and the table is not in 3NF. An "explain why the table is not in 3NF" answer names the dependency and the fields involved.



1NF removes the repeating group, 2NF the partial dependency, 3NF the transitive dependency

Worked example. A car-rental shop records each rental as `RENTAL(RentalID, RentalDate, CustomerID, CustomerName, CustomerPhone, CarReg, CarModel, DailyRate, Days)`, where one rental can include several cars. Explain why the table is not normalised and produce a 3NF design.

Not in 1NF: the car fields `CarReg, CarModel, DailyRate, Days` form a **repeating group** – one rental has several cars. Move them to `RENTAL_CAR(RentalID, Car`

`Reg, CarModel, DailyRate, Days)` with the composite key `(RentalID, CarReg)`. *Not in 2NF:* in `RENTAL_CAR`, `CarModel` and `DailyRate` depend on `CarReg` alone – a **partial dependency**. Move them to `CAR(CarReg, CarModel, DailyRate)`, leaving `RENTAL_CAR(RentalID, CarReg, Days)`. *Not in 3NF:* in `RENTAL`, `CustomerName` and `CustomerPhone` depend on `CustomerID`, a non-key field – a **transitive dependency**. Move them to `CUSTOMER(CustomerID, CustomerName, CustomerPhone)`, leaving `RENTAL(RentalID, RentalDate, CustomerID)`. The 3NF design is four tables – `CUSTOMER`, `RENTAL`, `RENTAL_CAR`, `CAR` – with `CustomerID`, `RentalID` and `CarReg` as foreign keys; underline every primary key.

Database Management System (DBMS)

A **DBMS** manages the database centrally. Features that fix the file-based limits:

- **data dictionary** – a description of every table, field, type and key; programs query it instead of hard-coding the structure.
- redundancy/consistency control – each fact stored once.
- **concurrent access** control – locks and transactions let many users work at once.
- **backup** and recovery; security and per-user permissions.
- **integrity rules** – keys, unique and range constraints, enforced centrally.
- **transactions** – a group of operations that all succeed or all fail.
- **views** – virtual tables that show each user "their" slice of the data.
- **data management** and **data modelling** – control how data is stored and define its structure as a **logical schema** (the logical design, independent of physical storage).
- **data integrity** and **data security** – enforce correctness and control access centrally.
- a **query processor** runs queries; a **developer interface** gives tools and APIs for building applications.

Its tools include a data-dictionary editor, a query builder, a forms builder, a report generator, user management, and an SQL editor.

What the data dictionary holds (a "give three items" question): the names of the tables; the names of the fields in each table; each field's data type and length; the primary and foreign keys and the relationships between tables; validation rules; indexes; and who may access each table. It is metadata – data *about* the data – and the DBMS uses it to check every query and every change.

How the DBMS keeps the data secure (a "describe two methods" question): **authentication** – a username and password, or a biometric, before any access; **access rights** – each user or group is allowed to read, write or delete only certain tables or fields, often through a **view**; **encryption** of the stored data and of

data sent to it, so a copied file is unreadable; **backups** taken regularly, so the data can be restored after loss; and a transaction log that records who changed what.

The two software tools. The **developer interface** is what a programmer uses to build the database and the applications on it: create tables and set keys and validation, write queries and SQL, and design forms and reports, without knowing how the data is physically stored. The **query processor** takes a query (SQL from a program, or a query built in the interface), checks it against the data dictionary, works out the most efficient way to run it, retrieves the data and returns the results.

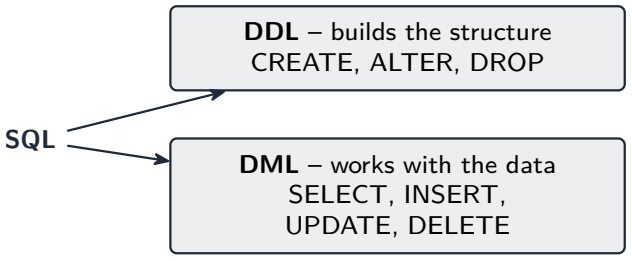
Logical schema. The DBMS keeps the **logical** design (which tables and fields exist and how they relate) separate from the **physical** storage (files, indexes, disk blocks). Programs work with the logical schema, so the physical storage can be reorganised without changing a single program – this is the data independence the file-based approach lacked.



The physical storage the logical schema hides: a hard disk's spinning platters and read/write head

DDL and DML

SQL (Structured Query Language) has two halves:



DDL builds the database structure; DML works with the data

- **Data Definition Language** (DDL) – creates or changes the **structure** (tables, keys, constraints).
- **Data Manipulation Language** (DML) – works with the **data** (insert, update, delete, **query**).

DDL basics

```
CREATE TABLE CUSTOMER (
  CustomerID INTEGER PRIMARY KEY,
  Name VARCHAR(50) NOT NULL,
  Phone VARCHAR(20)
);
```

Add a foreign key:

```
CREATE TABLE ORDER (
  OrderID INTEGER PRIMARY KEY,
  CustomerID INTEGER,
  OrderDate DATE,
  FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID)
);
```

Modify and drop:

```
ALTER TABLE CUSTOMER ADD Email VARCHAR(100);
DROP TABLE CUSTOMER;
```

Common types: `INTEGER`, `REAL`, `VARCHAR(n)`, `CHAR(n)` (also `CHARACTER(n)`), `DATE`, `TIME`, `BOOLEAN`, `DECIMAL(p, s)`.

DML basics

Query with `SELECT`:



A SELECT query returns only the rows that match its condition

```
SELECT Name, Phone
FROM CUSTOMER
WHERE City = 'London'
ORDER BY Name ASC;
```

`SELECT` lists fields, `FROM` names the table, `WHERE` filters rows, `ORDER BY` sorts.

A **join** combines two tables using a foreign-key relationship:

```
SELECT C.Name, O.OrderDate
FROM CUSTOMER C INNER JOIN ORDER O
  ON C.CustomerID = O.CustomerID
WHERE O.OrderDate >= '2024-01-01';
```

```
SELECT C.Name, COUNT(D.DeviceID) AS Devices ..... the fields to output; a calculated column, given a name
FROM CUSTOMER C ..... the first table, with a short alias
INNER JOIN DEVICE D ON C.CustomerID = D.CustomerID ..... the second table, linked through the foreign key
WHERE D.Type = 'phone' ..... keep only the rows that match
GROUP BY C.Name ..... one output row per customer
ORDER BY Devices DESC; ..... sort the result, largest first
```

The parts of a query, in the order they must be written

Aggregate functions (`COUNT`, `SUM`, `AVG`, `MIN`, `MAX`) are often used with `GROUP BY`:

```
SELECT CustomerID, COUNT(*) AS NumOrders
FROM ORDER
GROUP BY CustomerID;
```

Insert, update, delete:

```
INSERT INTO CUSTOMER (CustomerID, Name, Phone)
VALUES (101, 'Ada Lovelace', '020-1234-5678');

UPDATE CUSTOMER SET Phone = '020-9999-0000' WHERE CustomerID = 101;

DELETE FROM CUSTOMER WHERE CustomerID = 101;
```

Always put a `WHERE` clause on `UPDATE` and `DELETE`, or the change hits **every** row.

Tips for exam SQL

- use the exact table and field names from the question.
- quote strings with single quotes (`'Smith'`); don't quote numbers.
- comparisons: `=`, `<`, `>`, `<=`, `>=`, `<>`.
- `LIKE 'A%'` matches anything starting with A (`%` = any string, `_` = one character); `IN (1,2,3)`; `BETWEEN 10 AND 20`.
- combine conditions with `AND` / `OR` / `NOT`, and end each statement with a semicolon.

The DDL pattern the exam wants. Every `CREATE TABLE` names each field with its type, marks the primary key, and declares each foreign key with the table it references; a composite key is declared on its own line:

```
CREATE TABLE RENTAL_CAR (
  RentalID INTEGER,
  CarReg VARCHAR(8),
  Days INTEGER,
  PRIMARY KEY (RentalID, CarReg),
  FOREIGN KEY (RentalID) REFERENCES RENTAL(RentalID),
  FOREIGN KEY (CarReg) REFERENCES CAR(CarReg)
);
```

> **Worked example.** Using `CUSTOMER(CustomerID, Name, Phone)` and `DEVICE(DeviceID, CustomerID, Type, Model)`, write SQL scripts to: (a) list the name and phone number of every customer who owns a device of type `'tablet'`, in alphabetical order of name; (b) count the devices of each type; (c) record that customer 17 now has the phone number `'0771 234 5678'`; (d) add a new device, ID 305, a `'laptop'` of model `'X1'` belonging to customer 17.

(a)

```
SELECT CUSTOMER.Name, CUSTOMER.Phone
FROM CUSTOMER INNER JOIN DEVICE
  ON CUSTOMER.CustomerID = DEVICE.CustomerID
WHERE DEVICE.Type = 'tablet'
ORDER BY CUSTOMER.Name ASC;
```

(b)

```
SELECT Type, COUNT(DeviceID) AS NumberOfDevices
FROM DEVICE
GROUP BY Type;
```

(c) `UPDATE CUSTOMER SET Phone = '0771 234 5678' WHERE CustomerID = 17;`
(d) `INSERT INTO DEVICE (DeviceID, CustomerID, Type, Model) VALUES (305, 17, 'laptop', 'X1');`

Marks are given per clause – the fields, the tables, the join condition, the `WHERE`, the `ORDER BY` – so a script with one wrong clause still scores the rest. Write `Table.Field` whenever two tables are involved.

> **Worked example.** Explain what this script does: `SELECT T.Name, SUM(R.Cost) AS Total FROM TECHNICIAN T INNER JOIN REPAIR R ON T.TechnicianID = R.TechnicianID GROUP BY T.Name;`

It outputs each technician's name with the total cost of the repairs that technician has carried out, one row per technician: the two tables are joined on TechnicianID, the rows are grouped by name, and the costs in each group are added. When asked what a script does, describe the result, not the syntax.

SQL conditions and keys

`LIKE 'Li%'` matches text starting with `Li`; `%` represents any number of characters and `_` exactly one. `AND` requires both conditions; `OR` requires at least one. Use brackets when combining them.

```
CREATE DATABASE Club;
CREATE TABLE Booking (
  MemberID INTEGER,
  EventID INTEGER,
  PRIMARY KEY (MemberID, EventID)
);
ALTER TABLE Booking ADD FOREIGN KEY (MemberID)
  REFERENCES Member(MemberID);
SELECT Name FROM Member
WHERE Name LIKE 'Li%' AND (Year = 12 OR Year = 13)
ORDER BY Name DESC;
```

The composite primary key makes the **pair** unique, not each column separately. A foreign key prevents a non-null reference to a missing parent record. `ON UPDATE CASCADE` carries a parent-key change to referencing rows; `ON DELETE CASCADE` deletes those rows when the parent is deleted.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
entity	something about which data is stored – a person, object or event – which becomes a table in a relational database
attribute	one item of data about an entity (a column of the table)
tuple	one row of a table: one instance of the entity
primary key	an attribute, or combination of attributes, that uniquely identifies each record in a table
foreign key	an attribute in one table whose value matches a primary key in another table, used to link the two
candidate key	any attribute (or combination) that could be chosen as the primary key
secondary key	a non-primary attribute that is indexed so the table can be searched or sorted on it quickly
composite key	a primary key made of two or more attributes together

Term	Definition
referential integrity	every foreign-key value must match an existing primary-key value in the table it refers to
first normal form	a table in which every attribute is atomic, there are no repeating groups, and there is a primary key
second normal form	in 1NF, and every non-key attribute depends on the whole of the primary key (no partial dependency)
third normal form	in 2NF, and no non-key attribute depends on another non-key attribute (no transitive dependency)
data dictionary	the metadata a DBMS keeps about the structure of the database: tables, fields, types, keys, relationships, validation
DDL / DML	the language used to define or change the structure of a database / the language used to query and maintain the data in it

Exam tips

- Define the terms exactly: entity, attribute, **primary key, foreign key**, and the relationship types (1:1, 1:many, many:many).
- Give a reason at each normal form: **1NF** (no repeating groups), **2NF** (no partial dependency), **3NF** (no non-key dependency) – and name the fields involved.
- Explain what a **DBMS** provides (data independence, security, integrity, concurrent access, a data dictionary, a developer interface, a query processor).
- Distinguish **DDL** (define the structure) from **DML** (query and change the data), and write SQL clause by clause: `SELECT`, `FROM`, `INNER JOIN … ON`, `WHERE`, `GROUP BY`, `ORDER BY`.
- To draw an E-R diagram from tables, find each foreign key first: every foreign key is one one-to-many relationship, with the "many" at the table that holds it.

Common mistakes

- Drawing a many-to-many relationship directly. It must be split into two one-to-many relationships through a link table holding both foreign keys.
- Explaining "not in 3NF" by "the data is repeated". Name the dependency (partial or transitive) and the fields involved.
- Double quotes round strings in SQL, or quotes round numbers. Strings take `'single quotes'`; numbers take none.

- Leaving out the `ON` condition after `INNER JOIN`. Without it the two tables are not linked.
- Putting an ordinary field next to `COUNT` or `SUM` in a `SELECT` without a `GROUP BY`.
- `UPDATE` or `DELETE` without a `WHERE`. It changes or removes every row in the table.

Algorithm Design and Problem-solving

A-Level Computer Science

Computational thinking

Computational thinking is the set of mental tools for **analysing a problem and designing a solution a computer can run**. Two key ones are abstraction and decomposition.



Computational thinking breaks a big problem into smaller, easier parts – like solving a jigsaw

Abstraction

Abstraction means **keeping the essential features** of a problem and **ignoring the irrelevant detail**, giving a simpler model.

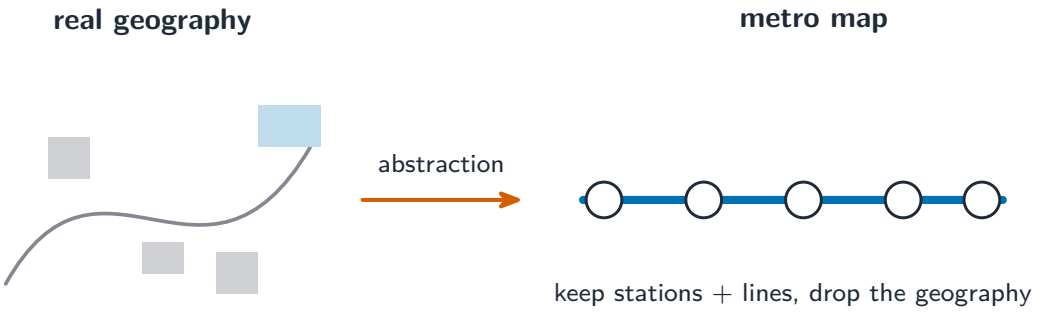
Examples:

- a **train-network map** keeps the stations and lines but drops the geography.
- a **class** in object-oriented programming keeps only the attributes and methods the system needs.
- a **function** hides a piece of work behind a name.

A full model of any real problem would be too big to reason about, so abstraction is essential.

The examiner asks for the **purpose** of abstraction and for its **benefits**. Purpose: to produce a simpler model of a problem that contains only the details needed to solve it. Benefits: the problem is easier to understand and to program; the program is smaller and faster to write and test; the same model can be reused for similar problems. When

you are asked to **produce an abstract model** of a system, list only the data and actions the task needs. For a school timetable that means the classes, rooms, teachers and periods; it does not mean the colour of the rooms or the age of the teachers.



Abstraction keeps the essentials (stations and lines) and drops irrelevant detail (the geography)

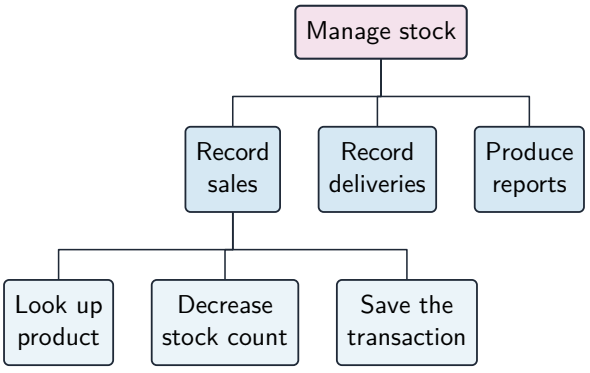
Decomposition

Decomposition means **breaking a large problem into smaller sub-problems**, each easier to solve and tackled one at a time.

1. find the main parts of the task.
2. break each into smaller sub-tasks.
3. continue until each is small enough to design directly.
4. solve the small tasks and combine them.

For stock control: "manage stock" → "record sales", "record deliveries", "produce reports" → ("record sales") "look up product", "decrease stock count", "save the transaction". Decomposition makes big problems manageable, lets a team divide the work, and gives modular code – each module becomes a **procedure** or function.

"Explain why decomposition is used" is a three-mark question with a fixed shape. Give three separate benefits: each **sub-problem** is small enough to design, code and test on its own; different programmers can work on different **modules** at the same time; a module that already exists (or a library routine) can be reused, and a fault is easier to find because it lies inside one module. A structure chart (topic 12) is the diagram of a decomposition: the program at the top, its modules beneath, and the data passed between them.



Decomposing a program into modules and sub-modules

Algorithms

An **algorithm** is a **solution expressed as a sequence of defined steps**. Each step is **unambiguous** (one meaning), **deterministic** (same input → same output), **finite** (the steps end), and **effective** (each can be done). An algorithm says *what to do*, independent of the programming language used to implement it.

Identifier table

When you start an algorithm, list every piece of data in an **identifier table** – its **identifier** (the **variable** name), **data type**, and description. The exam's table has exactly these three columns:

Identifier	Data type	Description
`Category`	`STRING`	the product category
`SaleDate`	`DATE`	when the item was sold
`ItemCost`	`REAL`	cost of the item
`InStock`	`BOOLEAN`	`TRUE` if in stock
`Sales`	`ARRAY[1:30] OF REAL`	the last 30 daily sales totals

Use **descriptive** names (`ItemCost`, not `x`): an identifier starts with a letter, contains no spaces, and is written the same way every time it appears. Common types are `INTEGER`, `REAL`, `STRING`, `CHAR`, `BOOLEAN`, `DATE`, plus arrays. The table forces you to name every piece of data before writing code, and a "complete the identifier table" question gives one mark for each correct data type or description, so write the type exactly as the pseudocode guide does.

list every piece of data first - name, type, description

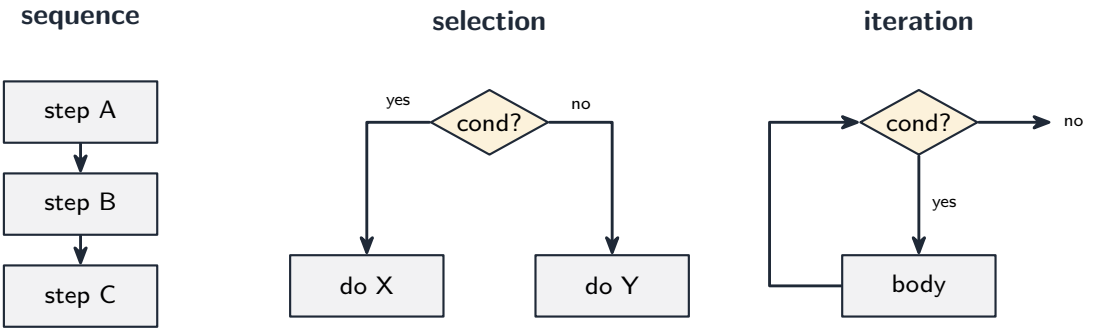
name	data type	description
`ItemCost`	`REAL`	cost of the item
`Category`	`STRING`	the product category
`SaleDate`	`DATE`	when the item was sold
`InStock`	`BOOLEAN`	TRUE if in stock

use **descriptive** names (`ItemCost`, not `x`) - the table forces you to plan the data

An identifier table names every piece of data before you write code

Pseudocode – the three basic constructs

Pseudocode is a structured, language-neutral way to describe algorithms.



The three building blocks of any algorithm: sequence, selection and iteration

1. Sequence

Steps run one after another (**sequence**):

```
INPUT Name
INPUT Age
OUTPUT "Hello", Name
```

2. Selection

A choice of which steps run, based on a condition (**selection**):

```
IF Age >= 18 THEN
    OUTPUT "Adult"
ELSE
    OUTPUT "Minor"
ENDIF
```

For more options, use `CASE OF ... ENDCASE`.

3. Iteration

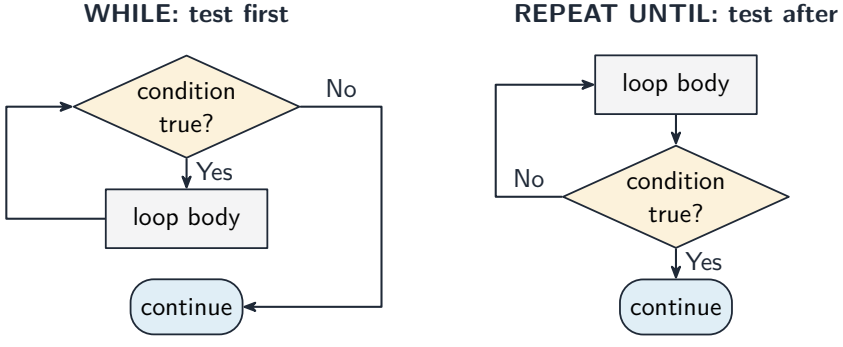
Repeating a block (**iteration** , a **loop**):

```
FOR i ← 1 TO 10
    OUTPUT i
NEXT i
```

A **WHILE** loop tests the condition **before** each pass (may run zero times); a **REPEAT...UNTIL** loop tests **after** each pass (always runs at least once).

```
WHILE Total < 100 DO
    INPUT Value
    Total ← Total + Value
ENDWHILE

REPEAT
    INPUT Mark
UNTIL Mark >= 0 AND Mark <= 100
```



A WHILE loop tests before the body runs; a REPEAT ... UNTIL loop tests after it, so its body always runs at least once

Choosing the loop is itself a mark: `FOR` when you know how many times (a **count-controlled loop**); `WHILE` when the loop might not run at all (a **pre-**

condition loop); `REPEAT ... UNTIL` when it must run at least once, as in validating an input (a **post-condition loop**). A "describe the iteration construct" answer names the construct, says where the condition is tested, and gives the consequence (zero times or at least once).

Common operations

- **assignment** : `x ← 5` (an arrow; `=` is for comparison).
- input/output: `INPUT variable`, `OUTPUT expression`.
- comparisons `=`, `<>`, `<`, `>`, `<=`, `>=`; logic `AND`, `OR`, `NOT`.
- arithmetic `+ - * /`, plus `DIV` (integer division) and `MOD` (remainder).
- strings: `LENGTH`, `LEFT`, `RIGHT`, `MID`, and `&` for **concatenation** (joining).

The pseudocode the exam expects

Every pseudocode answer is marked against Cambridge's published pseudocode guide. Write these forms exactly:

Construct	Pseudocode
Variable	`DECLARE Total : INTEGER`
Array	`DECLARE Marks : ARRAY[1:30] OF REAL`
Constant	`CONSTANT MaxTries = 3`
Assignment	`Total ← Total + Value`
Input / output	`INPUT Name` `OUTPUT "Hello ", Name`

Selection	`CASE OF Choice`
	`1 : OUTPUT "Add"`
	`OTHERWISE OUTPUT "Error"`
	`ENDCASE`

FOR loop	`FOR i ← 1 TO 10 STEP 2 ... NEXT i`
WHILE loop	`WHILE Total < 100 DO ... ENDWHILE`
REPEAT loop	`REPEAT ... UNTIL Mark >= 0`
Integer arithmetic	`17 DIV 5 = 3` `17 MOD 5 = 2`
Strings	`LENGTH(S)`, `LEFT(S, 3)`, `RIGHT(S, 2)` `MID(S, 2, 4)`, `TO_UPPER(S)`, `TO_LOWER(S)`
Conversions	`INT(3.7) = 3`, `NUM_TO_STR(12)` `STR_TO_NUM("4.5")`, `ASC('A') = 65`, `CHR(66) = 'B'`

Construct	Pseudocode
Random	`RAND(100)` `INT(RAND(100)) + 1`

`RAND(100)` gives a real number from 0 up to (but not including) 100. `INT(RAND(100)) + 1` gives an integer from 1 to 100.

Two habits earn marks on every question: **declare** every variable you use, with the type from your identifier table, and **initialise** every **counter** and total (`Count ← 0`, `Total ← 0`) before the loop that changes it.

■ Input → Process → Output

Every program follows this shape:

```
INPUT Length
INPUT Width
Area ← Length * Width
OUTPUT "Area = ", Area
```

Listing the inputs and outputs first makes the algorithm cleaner.

> **Worked example.** Write pseudocode that inputs 100 integers and outputs how many of them, and the total of those, that lie between 10 and 20 inclusive.

Identifier table: `Count : INTEGER` (loop counter), `Value : INTEGER` (the integer just input), `InRange : INTEGER` (how many were in range), `Total : INTEGER` (their sum).

```
DECLARE Count, Value, InRange, Total : INTEGER
InRange ← 0
Total ← 0
FOR Count ← 1 TO 100
    INPUT Value
    IF Value >= 10 AND Value <= 20 THEN
        InRange ← InRange + 1
        Total ← Total + Value
    ENDIF
NEXT Count
OUTPUT InRange, Total
```

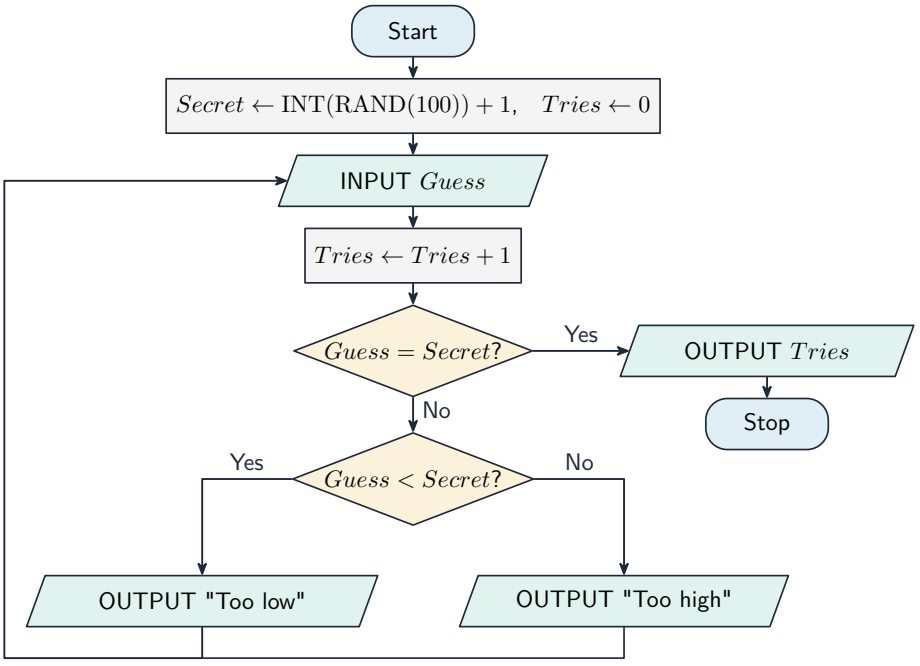
If the question then asks you to "identify two constructs and state how each is used", answer in the same shape: iteration, the `FOR` loop, repeats the input 100 times; selection, the `IF` statement, adds a value only when it is in range.

Worked example. A program picks a secret integer from 1 to 100. The user guesses until they are right; after each wrong guess the program says "Too low" or "Too high", and at the end it outputs how many guesses were made.

Identifier table: `Secret : INTEGER` (the number to guess), `Guess : INTEGER` (the user's input), `Tries : INTEGER` (how many guesses so far).

```
DECLARE Secret, Guess, Tries : INTEGER
Secret ← INT(RAND(100)) + 1
Tries ← 0
REPEAT
    INPUT Guess
    Tries ← Tries + 1
    IF Guess < Secret THEN
        OUTPUT "Too low"
    ELSE
        IF Guess > Secret THEN
            OUTPUT "Too high"
        ENDIF
    ENDIF
UNTIL Guess = Secret
OUTPUT "You took ", Tries, " guesses"
```

A `REPEAT ... UNTIL` loop is the right choice because the user must guess at least once. The marks are for: the random number in the right range, a loop that ends on a correct guess, the counter that starts at zero and increases inside the loop, the two messages under the right conditions, and the final output.

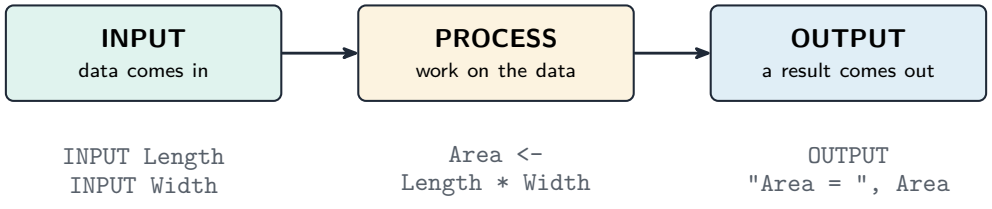


The same guessing game as a flowchart: the two decision diamonds are the two IF statements, and the return arrow is the REPEAT ... UNTIL loop

Worked example. Output two different random integers, each between −10 and 10 inclusive.

There are 21 possible values, so `INT(RAND(21))` gives 0 to 20 and subtracting 10 shifts it to the range −10 to 10. The second number must be generated again until it differs from the first:

```
DECLARE First, Second : INTEGER
First ← INT(RAND(21)) - 10
REPEAT
    Second ← INT(RAND(21)) - 10
UNTIL Second <> First
OUTPUT First, Second
```



Every program follows the Input, Process, Output shape

Three notations

The same algorithm can be written three ways.

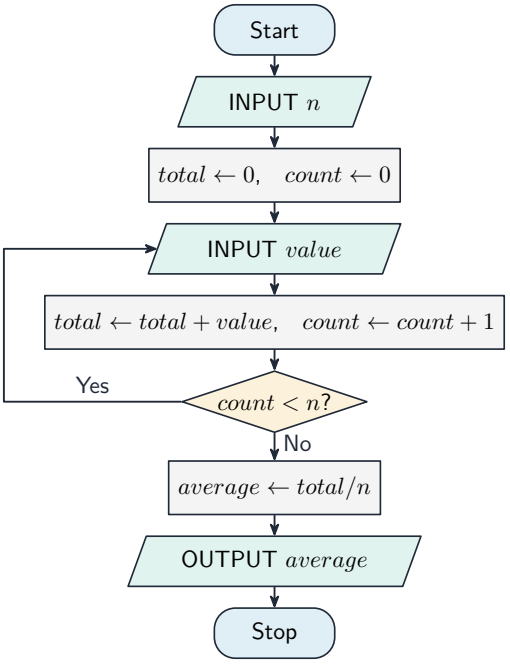
- **structured English** – natural language with indentation and fixed keywords; good for a high-level description.
- **flowchart** – a diagram with standard shapes:

Shape	Meaning
Rounded rectangle	Start / Stop
Parallelogram	Input / Output
Rectangle	Process
Diamond	Decision
Arrow	Flow of control

- **pseudocode** – the keyword notation above; closest to code.

You should be able to convert between any pair: each `IF` is a decision diamond, each loop is a back-arrow, and a sequence is stacked rectangles.

IF ... THEN ... ELSE ... ENDIF



A flowchart for averaging a list of numbers, using the standard shapes

Stepwise refinement

Stepwise refinement starts with a high-level outline and **expands each step** until it is small enough to code. For an average of n numbers:

Level 1:

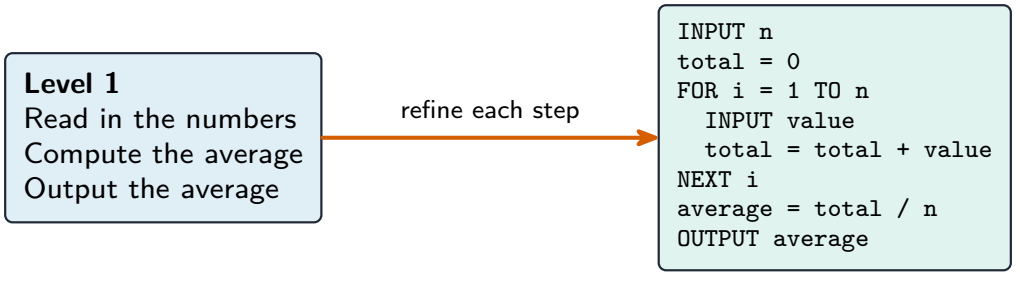
```
Read in the numbers
Compute the average
Output the average
```

Level 2:

```
INPUT n
total ← 0
FOR i ← 1 TO n
    INPUT value
    total ← total + value
NEXT i
average ← total / n
OUTPUT average
```

Each refinement keeps the previous structure and adds detail.

A six-mark "apply stepwise refinement" question gives you a high-level outline and wants each step expanded into the concrete statements a programmer could code. Keep the steps in the same order, name the data each step reads or produces, and stop when every line is a single input, assignment, output, loop or condition. For example, "validate the password" becomes: input the password; check its length is at least 8; check it contains at least one digit; output "accepted" if both checks pass, otherwise output "rejected".



Stepwise refinement: expand each high-level step into detailed pseudocode

Logic statements

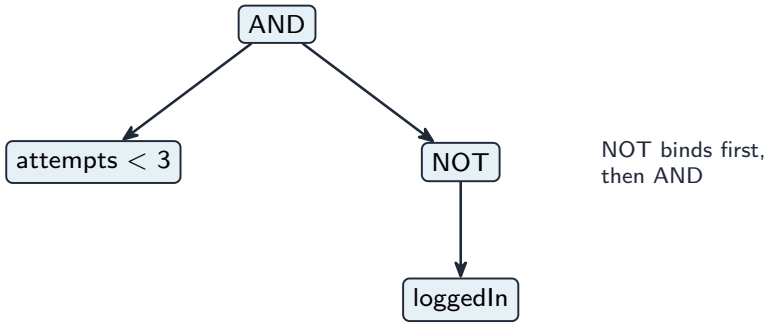
A **logic statement** is a **Boolean** condition that controls branching, built from comparisons (`x > 10`), connectives (`AND`, `OR`, `NOT`) and brackets. Use it as the condition of `IF`, `WHILE` or `REPEAT...UNTIL`:

```
WHILE attempts < 3 AND NOT loggedIn DO
    INPUT password
    IF password = correctPassword THEN
        loggedIn ← TRUE
    ELSE
        attempts ← attempts + 1
    ENDIF
ENDWHILE
```

Precedence (highest to lowest): `NOT`, then `AND`, then `OR`. Use brackets when unsure. Common mistakes:

- `a = 1 OR 2` is wrong – write `a = 1 OR a = 2`.
- `NOT a > 5` means `NOT (a > 5)`, i.e. `a <= 5`.
- `NOT (A AND B)` is the same as `(NOT A) OR (NOT B)` (**De Morgan's law**) – handy for simplifying conditions.

Turning a sentence into a logic statement is a skill the papers test directly. "A ticket is free for anyone under 5 or over 65" becomes `Age < 5 OR Age > 65`. "A mark is valid if it is a whole number from 0 to 100" becomes `Mark >= 0 AND Mark <= 100`. "The loop stops when the file is finished or ten records have been read" becomes `UNTIL EOF(File) OR Count = 10`. Write each comparison in full: `Age > 65` and `Age < 5`, never `Age > 65 OR < 5`.



Precedence: NOT binds to loggedIn first, then AND combines the two sides

Worked example. Write an identifier table and pseudocode to read 10 numbers and output the largest. The **identifier table** names each variable with its data type and purpose: `Count : INTEGER` (loop counter), `Num : REAL` (the number just read), `Max : REAL` (largest so far).

```
Max ← -999999
FOR Count ← 1 TO 10
    INPUT Num
    IF Num > Max THEN
        Max ← Num
    ENDIF
NEXT Count
OUTPUT Max
```

The design decision carrying the marks is **initialising `Max`**: it must start **lower than any possible input** - or, safer still, be set to the **first** number read. Initialise it to `0` and the algorithm wrongly returns 0 for a list of negative numbers, a bug your trace only exposes if the test data include a negative.

■ Programming prerequisites for algorithm questions

Before writing or tracing algorithms, use the expression and built-in-function methods on sheets 11.1.1–11.1.2. `TO_UPPER` and `TO_LOWER` accept whole strings; `UCASE` and `LCASE` in the guide act on a single character. Follow the question's insert.

Join strings with `&`; convert a number with `NUM_TO_STR` before joining it. Call a procedure with `CALL Name(arguments)`; use a function as an expression. `SETDATE(day, month, year)` constructs a DATE value. Check types before evaluation: an invalid expression is `ERROR`, not a guessed conversion.

For a flowchart with several totals, initialise **each** accumulator before the loop. Update them on the correct branches, advance the count, then test termination. Put averages after the loop and use the number of values actually included.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
abstraction	keeping the essential details of a problem and leaving out the details that are not needed
decomposition	breaking a problem down into smaller sub-problems, each of which can be solved separately
algorithm	a solution to a problem expressed as a sequence of defined steps

Term	Definition
identifier table	a table listing each identifier used in an algorithm with its data type and a description of its purpose
pseudocode	a structured, language-independent way of writing the steps of an algorithm
flowchart	a diagram that shows the steps and decisions of an algorithm using standard symbols joined by arrows
sequence	statements executed one after another in the order written
selection	choosing which statements to execute according to a condition
iteration	repeating a group of statements while, or until, a condition holds
stepwise refinement	breaking each step of an outline into smaller steps, repeatedly, until each step can be coded directly
logic statement	a condition built from comparisons and the operators AND, OR and NOT that evaluates to TRUE or FALSE

Exam tips

- Define an **algorithm** as an unambiguous, finite, deterministic sequence of steps, independent of language.
- Use the three constructs correctly – **sequence, selection, iteration** – and keep an identifier table with data types.
- Break a problem down by **decomposition and abstraction**, then stepwise refinement.
- Write pseudocode that would actually run: declare variables and follow the exam's pseudocode style.

Common mistakes

- Using `=` to assign a value. Assignment is `←`; `=` is a comparison.
- Forgetting `ENDIF`, `ENDWHILE`, `ENDCASE` or `NEXT`. Every construct closes, and the closing word is where the mark for the construct is checked.
- Not initialising a total or counter before the loop, so the algorithm adds to a value that never existed.
- Using a `FOR` loop when the number of repetitions is unknown. Reading until a sentinel value or a correct guess needs `WHILE` or `REPEAT ... UNTIL`.

- Writing `Age > 65 OR < 5`. Each side of `OR` and `AND` must be a complete comparison.
- Answering "explain why decomposition is used" with one benefit written three ways. Three marks need three different benefits.

Data Types and Structures

A-Level Computer Science

Choosing data types

Every variable needs a **data type** – the kind of value it holds and the operations allowed:

- `INTEGER` – a whole number (`42`, `-7`). For counts, indexes, IDs.
- `REAL` – a number with a fractional part (`3.14`). For money, measurements.
- `STRING` – characters in quotes (`"Hello"`). For text.
- `CHAR` – a single character (`'A'`).
- `BOOLEAN` – `TRUE` or `FALSE`. For flags.
- `DATE` – a calendar date.

Pick the smallest precise type that fits: `INTEGER` for whole counts, `BOOLEAN` for flags (not the strings `"yes"`/`"no"`).

The "give the appropriate data type" tables are decided by how the value is used: the average mark of a class is `REAL` (it has a fractional part); an email address is `STRING`; the number of students is `INTEGER`; whether a student has paid is `BOOLEAN`; a date of birth is `DATE`; an array index is always `INTEGER`; a single grade letter is `CHAR`; a phone number is a `STRING`, because it starts with `0` and is never used in arithmetic. A `BOOLEAN` is used for a flag with only two states: whether a search has found its target, whether a member has paid, whether a seat is booked. For the identifier table, the variable name must be meaningful too: `NumberOfPeople`, not `n`.

Records

A **record** (a **record structure**) holds **several fields of different types** under one name – useful when several values describe one thing.

```
TYPE TStockItem
    DECLARE ItemID : INTEGER
    DECLARE Category : STRING
    DECLARE ItemCost : REAL
    DECLARE InStock : BOOLEAN
ENDTYPE
```

This defines the **type** `TStockItem`; declare variables of it:

```
DECLARE Item1 : TStockItem
DECLARE Items : ARRAY[1:100] OF TStockItem
```

Use dot notation to reach each **field** :

```
Item1.Category ← "Fruit"
OUTPUT Item1.Category, " costs ", Item1.ItemCost
```

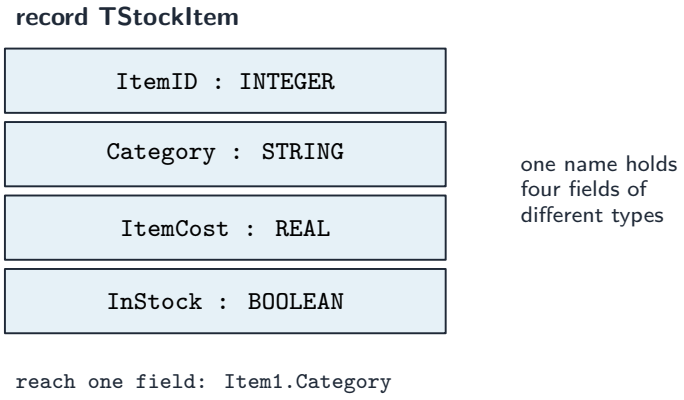
Use a record when values **always belong together** (a customer, a stock item); use separate variables for unrelated values.

Worked example. A club stores, for each student, a student ID (a string), a name, a date of birth and up to three club numbers (integers). Write pseudocode to declare the record type, an array to hold 3000 students, and a statement that stores a name in the first element.

```
TYPE Student
    DECLARE StudentID : STRING
    DECLARE Name : STRING
    DECLARE DateOfBirth : DATE
    DECLARE Club : ARRAY[1:3] OF INTEGER
ENDTYPE

DECLARE Membership : ARRAY[1:3000] OF Student
Membership[1].Name ← "Li Wei"
```

The marks: `TYPE` with the identifier and `ENDTYPE`; each field declared with a suitable type; the array declared with its bounds and `OF Student`; the field reached with the index and a dot. A ”state the error in the record declaration” question usually points at a missing `ENDTYPE`, a field with no type, or a field declared as a `STRING` that must hold arithmetic. Two conventions score marks on their own: an **unused element** is marked with a value that cannot be real data (an empty string, `-1`, an ID of `0`), and it is good practice to use the same marker everywhere so that every module can recognise an unused slot; an unused club field is `0`. The benefits of an array of records, for a ”state three benefits”: all the data for one entity is held under one identifier; the fields can have different data types; one array replaces several parallel arrays that would have to be kept in step; the whole set can be processed by one loop or passed as one parameter; and adding a field changes the type definition only. For one customer the suitable structure is a **record** (fields of different types under one name); for all customers it is an **array of records**.



A record holds several fields of different types under one name

Arrays

An **array** is an ordered collection of items of the **same type**, under one name, reached by an **index** .

- **element** – one item in the array.
- **bounds** – the lowest and highest valid indices.
- **dimension** – 1-D (a list), 2-D (a table), etc.
- **lower bound** and **upper bound** – the first and last valid index; the number of elements is upper bound minus lower bound plus one, and for a 2-D array the product of the two counts.

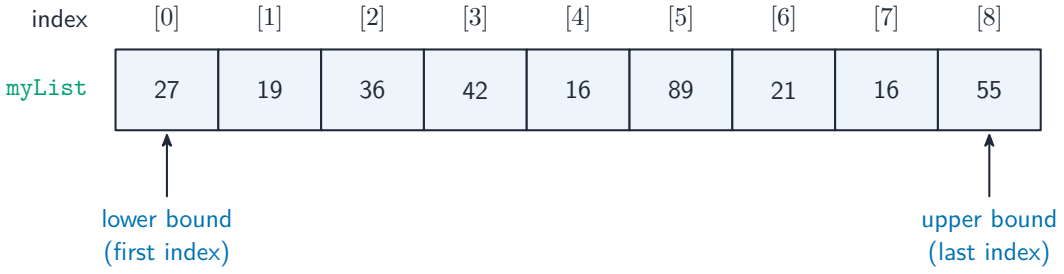
So in `ThisArray[n] ← 42` the array has one dimension, the index is the variable `n` (an `INTEGER`), and the element at that index receives `42`. Before an array can be declared you need its **data type** as well as its bounds. To declare 120 values that may include a decimal place: `DECLARE Data : ARRAY[1:120] OF REAL`; a 150-row, two-column table of strings: `DECLARE Data : ARRAY[1:150, 1:2] OF STRING`, which has 300 elements. The benefits of an array over separate variables, for a two-mark explain: one identifier instead of thirty; the elements can be processed by a loop with the index as the counter; the size is easy to change; and the whole set can be passed to a module as one parameter. An array can also replace a chain of selection statements: `Days InMonth[Month]` looks up the answer directly instead of twelve `IF` clauses, which is shorter, faster to write and easier to maintain.

1-D arrays

```
DECLARE Names : ARRAY[1:5] OF STRING
Names[3] ← "Cara"
OUTPUT Names[3]
```

Process every element with a `FOR` loop:

```
FOR i ← 1 TO 5
    OUTPUT Names[i]
NEXT i
```

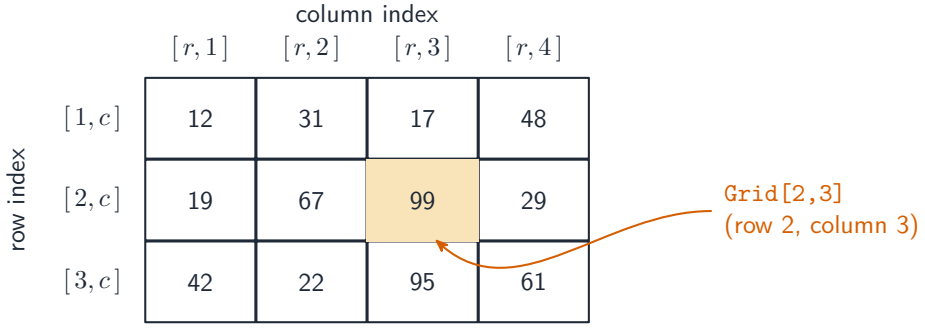


A 1-D array (a list) with indices and bounds

■ 2-D arrays (2D array)

```
DECLARE Grid : ARRAY[1:3, 1:4] OF INTEGER
Grid[2, 3] ← 99
```

The first index is the row, the second the column. Use nested loops to visit every cell. Use **1-D** for a single sequence, **2-D** for two natural dimensions (a grid, rows × columns).



A 2-D array (a table) with row and column indices

Common operations

A **linear search** checks each element until found:

```
FOR i ← 1 TO n
    IF A[i] = Target THEN
        OUTPUT "Found at ", i
    ENDIF
NEXT i
```

To find a sum, count, maximum or minimum, set a running variable then sweep through:

```
Max ← A[1]
FOR i ← 2 TO n
    IF A[i] > Max THEN
        Max ← A[i]
    ENDIF
NEXT i
```

A **bubble sort** puts an array in order: pass through it comparing each adjacent pair and **swapping** any that are out of order; repeat the passes until one pass makes no swaps.

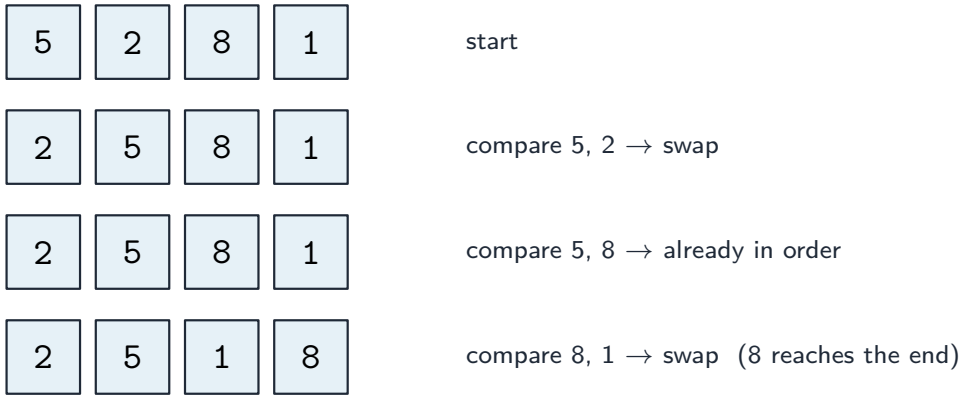
Paper 2 asks for these algorithms both as pseudocode and as **steps in words**, and sometimes in their "efficient" form:

- **Largest value**: set `Largest` to the first element; for each remaining element, if it is bigger than `Largest`, store it in `Largest`; after the loop output `Largest`. For the *position* of the largest, keep a second variable that stores the index each time `Largest` changes.
- **Linear search returning a position**: set `FoundAt ← -1` before the loop (a value that can never be a valid index, so it means "not found"); loop through the array; when the element matches, store the index and leave the loop; after the loop test `FoundAt`.
- **Count or output the non-blank elements**: compare each element with the marker for an unused element (`""` or `-1`) and count or output only those that differ.
- **Remove an item**: find its index by a linear search; move every later element one place towards the start, so the gap closes; mark the last element as unused (or decrease the count).
- **Insert into a sorted array**: find the first index whose element is larger; move that element and every later one one place towards the end; store the new value in the gap.

- **Efficient bubble sort**: a `Swapped` flag so that the passes stop as soon as a pass makes no swap, and an upper limit that falls by one each pass because the largest value has already reached the end.

```
REPEAT
    Swapped ← FALSE
    FOR Index ← 1 TO Limit - 1
        IF Data[Index] > Data[Index + 1] THEN
            Temp ← Data[Index]
            Data[Index] ← Data[Index + 1]
            Data[Index + 1] ← Temp
            Swapped ← TRUE
        ENDIF
    NEXT Index
    Limit ← Limit - 1
UNTIL Swapped = FALSE
```

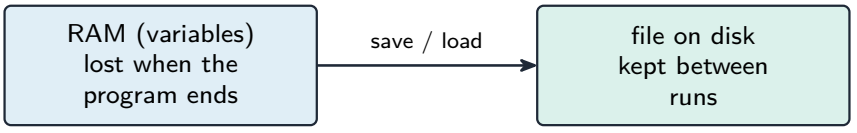
The marks are for the outer loop that repeats until no swaps, the flag set inside the `IF`, the three-line swap with a temporary variable, and the shrinking limit. A sort in "steps" (stepwise refinement) is: repeat until sorted; on each pass compare adjacent pairs; swap a pair that is out of order; after each pass the largest unsorted value is at the end. Two 1-D arrays of records or of parallel data are processed with one loop and one index; a 2-D array needs a nested loop, the outer over rows and the inner over columns, and a search in one row fixes the row index and loops over the column.



One pass of a bubble sort: adjacent pairs are compared and swapped, bubbling the largest value to the end

Files

A **file** is data stored on **secondary storage**, kept between program runs. Variables in RAM disappear when the program ends, so to save data permanently (high scores, records, settings) the program writes to a file. Files also let programs share data and restart from a saved state.



Variables in RAM vanish when the program ends; a file on disk persists between runs

A **text file** holds one or more lines of readable characters; programs read and write **text files** line by line. Open a file before use and **close** it after:

```
OPENFILE "data.txt" FOR READ      // or FOR WRITE, FOR APPEND
WHILE NOT EOF("data.txt") DO
    READFILE "data.txt", LineString
    OUTPUT LineString
ENDWHILE
CLOSEFILE "data.txt"
```

`EOF` tests the **end of file** before reading. To write:

```
OPENFILE "log.txt" FOR WRITE
FOR i ← 1 TO 100
    WRITEFILE "log.txt", "Event " & i
NEXT i
CLOSEFILE "log.txt"
```

Always **close** every file – otherwise buffered writes may be lost and other programs may be locked out.

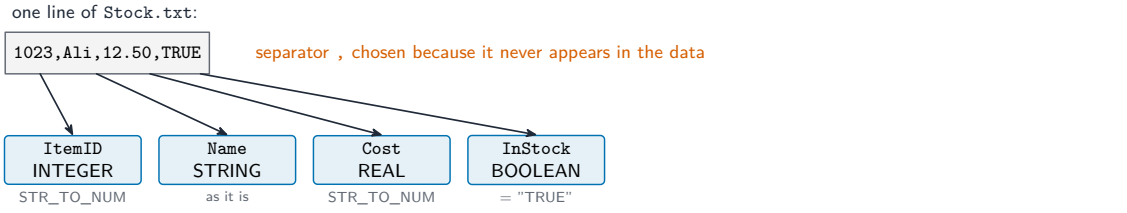
Why files (two marks): the data is kept after the program ends, so it is available the next time the program runs; it can be shared with other programs; and it can hold more than fits in memory. The characteristic of a text file that lets a program work through it is that it is a sequence of lines, read one after another from the start. The three **modes**: `READ` to read from the start; `WRITE` to create a new file, which **deletes any existing contents**, so it cannot be used to add to a file; `APPEND` to add lines at the end of an existing file. Test `EOF` before every read, and open the file only once, even when several modules use it.

Worked example. Write pseudocode for a procedure `LastLines(FileName : STRING)` that outputs the last three lines of a text file, in order.

```
PROCEDURE LastLines(BYVAL FileName : STRING)
    DECLARE LineX, LineY, LineZ : STRING
    LineX ← ""
    LineY ← ""
    LineZ ← ""
    OPENFILE FileName FOR READ
    WHILE NOT EOF(FileName) DO
        LineX ← LineY
        LineY ← LineZ
        READFILE FileName, LineZ
    ENDWHILE
    CLOSEFILE FileName
    OUTPUT LineX
    OUTPUT LineY
    OUTPUT LineZ
ENDPROCEDURE
```

Each new line pushes the previous three along, so when the file ends the three variables hold its last three lines; a file with fewer lines outputs empty strings. To output the *first* five lines, count the lines read and stop the loop at five or at `EOF`, whichever comes first; a file that is empty is detected by `EOF` being `TRUE` immediately after opening.

Fields in a line. A text file holds strings, so a record is written as one line with its fields joined by a **separator** character, and each number or Boolean converted with `NUM_TO_STR` (and read back with `STR_TO_NUM`, or by comparing with `"TRUE"`). Choose a separator that can never appear in the data: a comma or `|` for names and numbers, never a space when a name may contain one. If a field may contain any character, the separator can be confused with data; the fix is to put each field on its own line, or to write the field's length before it. One item per line is simple to read back but uses more lines and makes a record harder to see as a unit. Reading a file whose lines are in a known order (ascending by an ID) allows the search to stop as soon as a larger ID is read, instead of reading to the end. A save file that is created each time the game is saved needs a **meaningful filename**, for instance the player's name and the date and time, so that any earlier save can be restored.



One line of a text file is one record: fields joined by a separator, converted to their types when read back

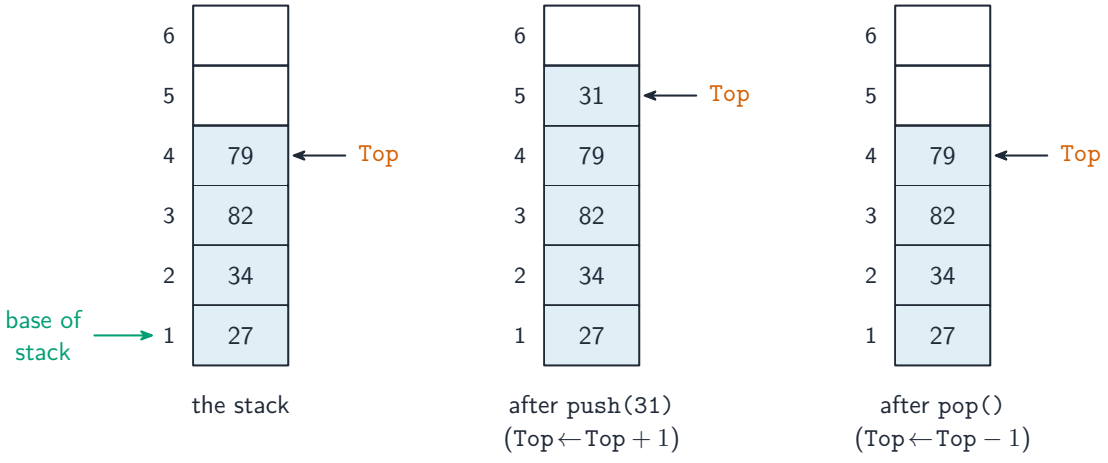
Abstract Data Types (ADTs)

An **Abstract Data Type** (ADT) is a **collection of data plus operations on it**, defined by **what** it does, not how it is stored. The user works only through the operations; the implementation is hidden, so it can change without affecting code that uses the ADT. Know three: stack, queue, linked list.

The one-mark definition: an ADT is a collection of data together with a set of operations on that data. A stack, a queue, a linked list, a binary tree and an array are all ADTs. To **justify** a choice: a queue when items must be handled in the order they arrived (print jobs, key presses, customers in a shop), because it is first in, first out; a stack when the most recent item must be handled first (undo, going back through web pages, reversing an order, the return addresses of nested calls), because it is last in, first out; a linked list when items are inserted and deleted in the middle of an ordered sequence often, because only pointers change and nothing has to be shifted. To **compare a stack and a queue**: both are linear structures of items with an order, both are implemented with an array and pointers, and both need a check for full before adding and for empty before removing; a stack has one pointer and adds and removes at the same end, a queue has two pointers and adds at one end and removes at the other.

■ Stack

A **stack** works in **LIFO** order (Last In, First Out). Operations: **push** (add to the top), **pop** (remove from the top), peek (look at the top), and tests for empty/full. Uses: undo history, function-call return addresses, expression parsing, backtracking.



Push and pop change the top pointer; the base pointer stays put

> **Worked example.** A stack of characters holds, from the bottom, `'P'`, `'N'`, `'Z'`, `'X'`, `'Y'`, `'W'`, with the top-of-stack pointer at `'W'` (memory location 202 of 200–207). The operations `POP`, `POP`, `PUSH 'A'`, `PUSH 'B'`, `POP` are performed. What is on the stack, and where does the pointer point?

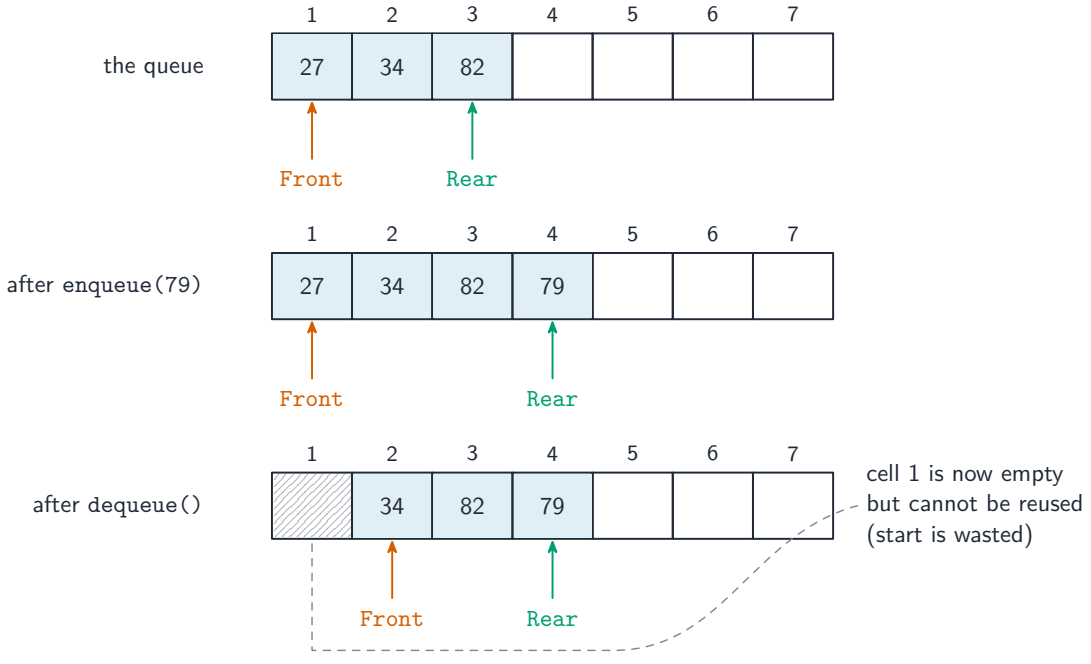
The two pops remove `'W'` then `'Y'`; the pushes add `'A'` then `'B'` in their places; the last pop removes `'B'`. The stack now holds `'P'`, `'N'`, `'Z'`, `'X'`, `'A'` and the pointer is at `'A'`, location 203. The value that has been on the stack longest is the bottom item, `'P'`; at most five further pops are possible before the stack is empty, and a pop on an empty stack is an error, which is why `Pop()` tests for empty first. A `Push()` function that returns `TRUE` on success first tests whether the pointer is at the top of the array (full) and returns `FALSE` if so. The array elements need no initialising before use, because the pointer alone says which elements are in use.



*A pile of books is a stack you can see. You can only add or take a book from the **top**, so the last one you put on is the first one you take off – that is exactly LIFO*

Queue

A **queue** works in **FIFO** order (First In, First Out). Operations: **enqueue** (add to the rear), **dequeue** (remove from the front), and tests for empty/full. Uses: print spooling, scheduling, breadth-first search, buffering.



Enqueue adds at the rear; dequeue removes from the front

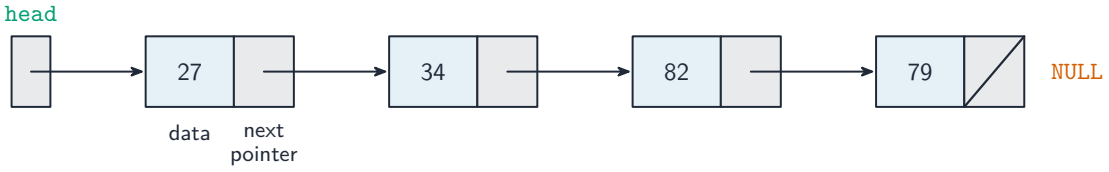
To **describe adding** an item: check that the queue is not full; store the item at the position given by the end-of-queue pointer; increment the end pointer (and the count). To **describe removing**: check that the queue is not empty; read the item at the front pointer; increment the front pointer (and decrement the count). State the convention you use: if the end pointer marks the next free space, front and end pointers being equal means the queue is **empty**; if it marks the last item, equal pointers mean one item. In a linear queue the front pointer only ever moves forward, so cells behind it are wasted; that is what the circular queue below fixes. The two features of a queue to state: items are added at the rear and removed from the front, so the first item added is the first removed.



*A line of people is a queue you can see. You join at the **back** and are served from the **front**, so whoever waited longest is served first – that is exactly FIFO*

Linked list

A **linked list** stores data as a sequence of **nodes**. Each node holds a value and a **pointer** to the next node; a head pointer marks the start, and the last node's pointer is a sentinel (e.g. `NULL`). Operations: insert, delete, search, and **traverse** (visit each node in order). Its advantage over an array is cheap insertion/deletion (just adjust pointers); its disadvantage is slow random access (you must follow pointers from the head).



A linked list: each node points to the next

Adding a node in order (four marks): traverse the list from the head, following the pointers, until the node before the position is found (the last node whose value is smaller); take a free node and store the new value in it; set the new node's pointer to the address the previous node pointed to; set the previous node's pointer to the new node. If the new value belongs at the front, the head pointer is changed instead. **Deleting** a node: find the node before it, and set that node's pointer to the address the deleted node pointed to, so the list bypasses it; the freed node returns to the free list. Compared with a 1-D array, inserting or deleting in a linked list needs no shifting of the other items, and the list can grow until memory runs out; the cost is the extra pointer stored with every item, and that reaching the nth item means following n pointers, since there is no direct index.

Implementing ADTs using arrays

Stack using an array

Hold items in `Stack[1:MaxSize]` with an integer `Top` (0 when empty).

- `Push(x)`: if `Top = MaxSize` the stack is full (**overflow**); else `Top ← Top + 1`; `Stack[Top] ← x`.
- `Pop()`: if `Top = 0` the stack is empty (**underflow**); else return `Stack[Top]` and `Top ← Top - 1`.

Queue using a circular array

A simple queue lets `Front` and `Rear` march off the end, wasting the start. The fix is a **circular array** – when a pointer reaches `MaxSize` it wraps back to 1:

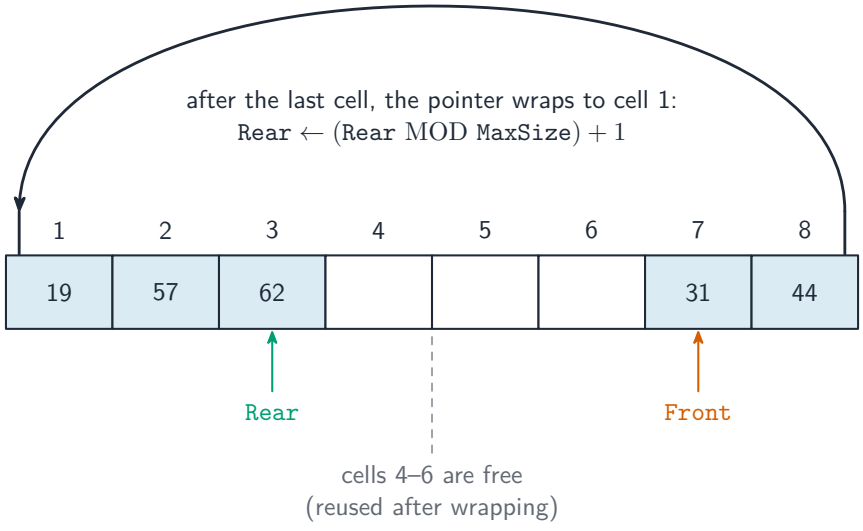
- `Enqueue(x)`: check full; else `Rear ← (Rear MOD MaxSize) + 1`; `Queue[Rear] ← x`.

- `Dequeue()`: check empty; else return `Queue[Front]` and `Front ← (Front MOD MaxSize) + 1`.

Track a separate count to tell empty from full.

The algorithm for the end pointer, in words: if the count equals the size, report that the queue is full and stop; otherwise add one to the end pointer; if it is now past the last index, set it to the first index; store the item there and add one to the count. The declarations that a five-mark "describe the declaration and initialisation" answer lists: the array with its size and element type; a front pointer and an end pointer, both initialised to the first index (or the front to the first index and the end to the next free space); and a count of items, initialised to 0.

For example, with `MaxSize = 6`: if `Rear = 5`, then `(5 MOD 6) + 1 = 6`, so the next item goes in cell 6; if `Rear = 6`, then `(6 MOD 6) + 1 = 1`, so the pointer **wraps back** to cell 1.



A circular queue wraps the pointers back to the start of the array

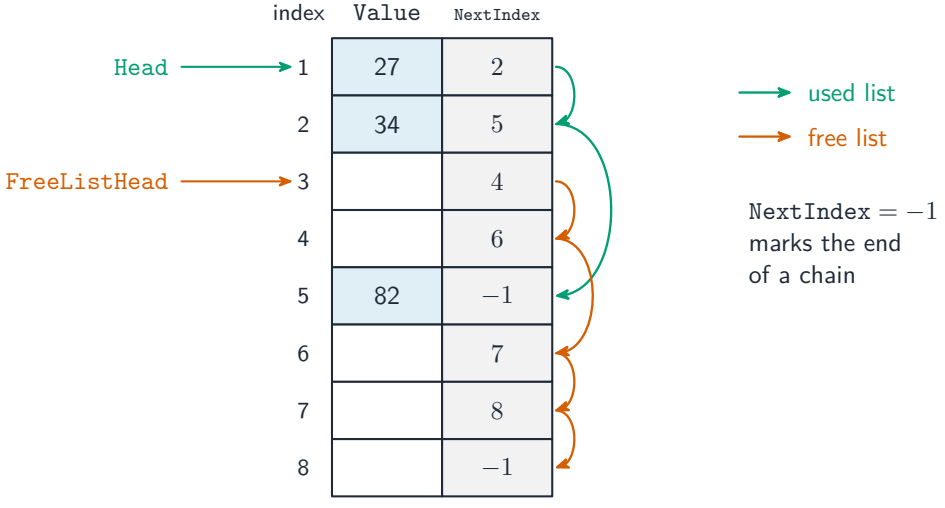
Linked list using an array

Use an array of records, each with a `NextIndex` index:

```
TYPE TNode
    DECLARE Value : INTEGER
    DECLARE NextIndex : INTEGER     // index of the next node, or -1 for
    ↪  end
ENDTYPE

DECLARE Nodes : ARRAY[1:MaxSize] OF TNode
DECLARE Head : INTEGER          // index of first node, -1 if empty
DECLARE FreeListHead : INTEGER // first available free node
```

A **free list** chains the unused slots, just as the data list chains its used ones. To insert: take a slot from `FreeListHead`, set the new node's value and `NextIndex`, and update the previous node's `NextIndex` (or `Head`). To delete: unlink the node and return its slot to the free list. This gives the flexibility of a linked structure with the static allocation of an array.



A linked list stored in an array: a data array and a pointer array

> **Worked example.** A linked list is held in a `Data` array and a `Pointer` array, with `Start` pointing to index 1. The list is $1 \to 3 \to 4$ (index 1 holds `D40`, index 3 holds `D32`, index 4 holds `D11`, whose pointer is $\emptyset$); the free list starts at index 2 and continues $2 \to 5$. Insert `D6` between `D32` and `D11`.

Take the first free node, index 2, and set `FreeStart` to its pointer, 5; store `D6` in `Data[2]`; set `Pointer[2]` to the value `Pointer[3]` held, which is 4; set `Pointer[3]` to 2. The list reads $1 \to 3 \to 2 \to 4$ and the free list is $5 \to \emptyset$. The answer to "how can the linked list be implemented" is exactly these parts: an array (or array of records)

for the data, a parallel array for the pointers holding indices, a start pointer, a free-list pointer and a null value such as -1 for the end.

> **Worked example.** A circular queue is held in an array of size 5 (indices 0 to 4) with `Front = 3`, `Rear = 3` and one item stored. Two items are added, then two are removed. Where are the pointers, and why use a *circular* queue at all? Every move uses `(pointer + 1) MOD size`, so the pointers **wrap**. Adding twice moves `Rear`: $3 \to 4$, then $4 \to 0$ (because $(4 + 1) \bmod 5 = 0$), so `Rear = 0` and three items are stored. Removing twice moves `Front` the same way: $3 \to 4$, then $4 \to 0$, leaving `Front = 0` and one item. The wrap is the whole point: in a **linear** array queue the pointers march to the end and the freed space at the front is wasted even when the queue is empty. Remember a queue removes at the **Front** and adds at the **Rear** - a stack uses one pointer for both.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly.

Term	Definition
record	a data structure that holds a set of data items (fields) of different data types under one identifier
array	a data structure that holds a fixed number of elements of the same data type under one identifier, each accessed by an index
index	the number that identifies one element of an array
upper bound, lower bound	the largest and smallest valid index of an array
text file	a file that stores data as lines of characters, which a program reads and writes one line at a time
abstract data type	a collection of data together with a set of operations on that data
stack	a list in which items are added to and removed from the same end, the top, so the last item added is the first removed (LIFO)
queue	a list in which items are added at the rear and removed from the front, so the first item added is the first removed (FIFO)
linked list	a list in which each node holds a data item and a pointer to the next node, with a start pointer to the first node

Term	Definition
pointer	a variable that holds the address (or index) of a node or of a position in a structure
linear search	checking each element in turn from the first until the target is found or the end is reached
bubble sort	repeated passes through the array comparing adjacent pairs and swapping those out of order, until a pass makes no swaps

Exam tips

- Choose the right **data structure** and justify it (a record for mixed fields, a 2-D array for a grid).
- Know how to implement a **stack, queue and linked list** with an array and pointers (top; front/rear; next).
- Distinguish an **ADT** (its behaviour) from its implementation (array plus pointers).

Common mistakes

- A record declaration without `ENDTYPE`, or fields without types. Every field is a `DECLARE` line with a type.
- Reading past the end of a file, or writing with `WRITE` when the file must keep its contents. Test `EOF` before each read; use `APPEND` to add.
- Writing a number to a text file without converting it. A file holds strings: `NUM_TO_STR` out, `STR_TO_NUM` back.
- Forgetting the checks. `Push` and enqueue test for full first; `Pop` and dequeue test for empty first, and the answer says so.
- Losing the rest of the list when inserting a node. Set the new node's pointer to the old next node before changing the previous node's pointer.
- A linear search that never says "not found". Initialise the position to -1 and test it after the loop.

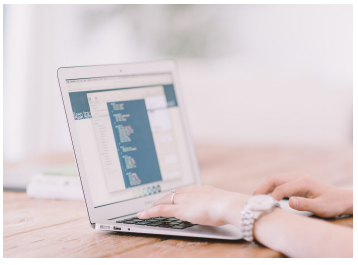
Programming

A-Level Computer Science

Programming basics

```
1 usage
public Hotel (Integer capacity) { this.capacity = capacity; }
1 usage
public synchronized boolean checkIn(AccommodationRequest request) {
    if (guests.size() < capacity) {
        guests.add(request);
        return true;
    }
    else {
        return false;
    }
}
```

Programming turns a design into instructions written as code

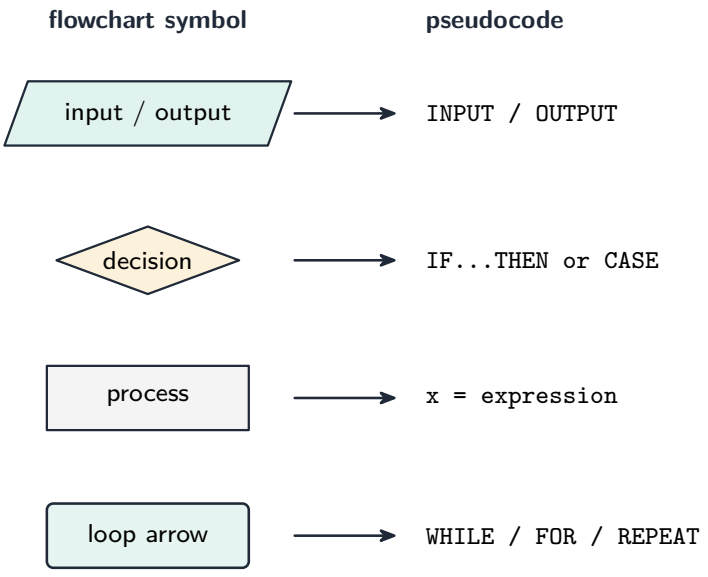


A programmer writes the code and tests it as they go

From design to code

You should be able to turn a design – a **flowchart** (**program flowchart**) or **structured English** – into **pseudocode**, and then into a real language:

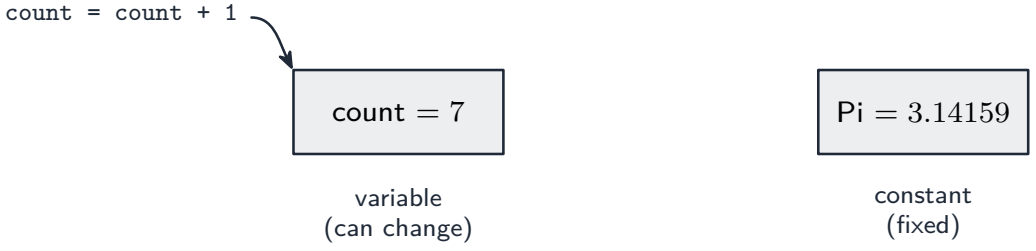
1. find the **variables** and their **data types**.
2. turn input/output boxes into `INPUT` / `OUTPUT`.
3. turn decision diamonds into `IF...ELSE...ENDIF` (or `CASE`).
4. turn loop arrows into `WHILE`, `REPEAT...UNTIL`, or `FOR`.
5. turn process boxes into assignments or calculations.
6. check by tracing a small input.



Each flowchart symbol becomes a pseudocode keyword

Constants and variables

A **constant** holds a value that **never changes**; a variable holds one that may change. Declare them with a type:



A variable's value can change; a constant stays fixed

```
CONSTANT Pi = 3.14159
DECLARE Radius : REAL
DECLARE Area : REAL

Radius ← 5
Area ← Pi * Radius * Radius
```

Use constants for fixed values that recur (`Pi`, `MaxScore`); they make code clearer and easy to change in one place.

In the exam, a constant is the answer to "identify a more appropriate way of representing" a fixed value, such as a tax rate or a maximum score, that appears at several places in the pseudocode. The benefits the scheme lists: the value is set once and **cannot be changed accidentally** by the program; a change is made **in one place** and reaches every statement that uses it; the identifier gives the value a **meaning** (`MaxScore` rather than `100`), so the code is easier to read and to check; and there is less risk of a typing error in a long value such as `3.14159`. A "state a value that could be replaced by a constant" question wants the literal from the pseudocode (`0.2`, `40`), not a new name.

Every variable is declared once, with an **identifier** (its name) and a data type, before it is used. The six types in the 9618 pseudocode guide:

Type	Holds	Written in the code as	Typical use
`INTEGER`	whole numbers	`42`, `-3`	a count, an array index, a loop counter
`REAL`	numbers with a fractional part	`3.75`	a price, an average
`CHAR`	one character	`'A'` (single quotes)	a grade letter, a menu key
`STRING`	a sequence of characters	`"Hello"` (double quotes)	a name, a postcode
`BOOLEAN`	`TRUE` or `FALSE`	`TRUE`	a flag such as `Found`
`DATE`	a calendar date	`12/05/2026`	a date of birth

A "give the appropriate data type" question is answered from how the variable is **used** in the pseudocode: a value with a decimal point is `REAL`; something set to `TRUE` or `FALSE` is `BOOLEAN`; a value in single quotes is `CHAR`; a value used as an array index, or with `DIV` and `MOD`, is `INTEGER`. Write the type in capitals, spelled as the guide spells it.

Worked example. State the appropriate data type for each variable.

```
Found ← FALSE
Initial ← 'K'
Price ← 12.99
Count ← Count + 1
Name ← "Li Wei"
```

`Found` is `BOOLEAN` (it holds `FALSE`); `Initial` is `CHAR` (one character in single quotes); `Price` is `REAL` (a decimal value); `Count` is `INTEGER` (a counter that goes up by one); `Name` is `STRING` (text in double quotes).

Assignment and expressions

Use ← for **assignment** :

```
Total ← Total + 1
Average ← Sum / Count
```

Expressions use **operators** :

- arithmetic `+ - * /`, plus `DIV` (integer division) and `MOD` (remainder): `7 DIV 2 = 3`; `7 MOD 2 = 1`.
- comparisons `=`, `<>`, `<`, `>`, `<=`, `>=`.
- logic `AND`, `OR`, `NOT`.

Precedence (highest to lowest): `NOT` → `* / DIV MOD` → `+ -` → comparisons → `AND` → `OR`. Use brackets when unsure.

Input and output

```
OUTPUT "Enter your name:"
INPUT Name
OUTPUT "Hello, ", Name
```

Built-in functions and library routines

Many tasks have ready-made **library routines** , so you need not write them. The Paper 2 **insert** lists the ones you may use, with their exact names, parameters and return types; any other function a question needs is given in the question. The names below are the insert's names. `VAL` and `STR` are IGCSE names and appear in neither 9618 document, so they earn nothing. `UCASE` and `LCASE` are a different case: they **are** 9618, defined in the Pseudocode Guide, but they take a single **`CHAR`**, and the insert does not list them at all – for a whole string on Paper 2 the routine is `TO_UPPER`.

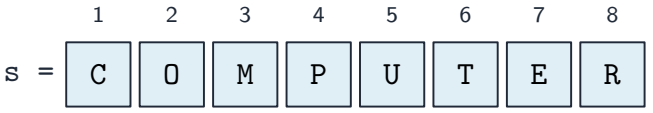
A **program library** holds routines that have already been written, compiled and tested; a program calls them instead of writing its own. The benefits the scheme accepts, for a "state three benefits" question: the routines are **already tested**, so they are less likely to contain errors; they **save development time**; they may do things the programmer could not write (complex statistics, graphics); they are written by experts and **reused** across many programs; and a routine with a fixed interface can be called from anywhere in the program.

Routine	Returns	Example
`LENGTH(s)`	the number of characters in `s`	`LENGTH("Hello") = 5`
`LEFT(s, n)` / `RIGHT(s, n)`	the first / last `n` characters	`RIGHT("Hello", 2) = "lo"`

Routine	Returns	Example
`MID(s, start, n)`	`n` characters from position `start` (positions count from 1)	`MID("Hello", 2, 3) = "ell"`
`TO_UPPER(s)` / `TO_LOWER(s)`	`s` in capitals / in small letters	`TO_UPPER("ab1") = "AB1"`
`NUM_TO_STR(x)` / `STR_TO_NUM(s)`	a number as a string / a string as a number	`STR_TO_NUM("3.5") = 3.5`
`IS_NUM(s)`	`TRUE` if `s` is a valid number	`IS_NUM("12a") = FALSE`
`ASC(c)` / `CHR(n)`	the character code of `c` / the character with code `n`	`ASC('A') = 65`, `CHR(66) = 'B'`
`INT(x)`	the whole-number part of `x`	`INT(7.9) = 7`
`RAND(n)`	a random real number from 0 up to, but not including, `n`	`INT(RAND(6)) + 1` is a dice roll
`DAY(d)`, `MONTH(d)`, `YEAR(d)`	the parts of a `DATE`	`YEAR(TODAY())`
`DAYINDEX(d)`, `SETDATE(d, m, y)`, `TODAY()`	the day of the week (1 = Sunday); a date built from three integers; today's date	
`EOF(f)`	`TRUE` when the file `f` has no more lines to read	`WHILE NOT EOF("data.txt")`

Strings are joined with `&` (**concatenation**): `"A" & "BC"` is `"ABC"`. Use the exact names from the insert, with the parameters in its order.

Dates and random numbers come up as one-line statements. `SETDATE(17, 11, 2007)` builds 17 November 2007; `12 - MONTH(MyDOB)` is the number of months from the month of birth to the end of the year; `IF DAYINDEX(MyDOB) = 5 THEN` tests for a Thursday, because Sunday is day 1. `RAND(n)` returns a real number from `0` up to, but not including, `n`, so a random **integer** from `Low` to `High` inclusive is `INT(RAND(High - Low + 1)) + Low`: `INT(RAND(21)) - 10` gives a value from `-10` to `10`.




```
LENGTH(s) = 8              RIGHT(s, 2) = "ER"
LEFT(s, 3) = "COM"         TO_UPPER(s) = "COMPUTER"
MID(s, 4, 3) = "PUT"       TO_LOWER(s) = "computer"
```

The common string routines acting on `s = "COMPUTER"` (positions 1–8)

Worked example. Evaluate each expression, given `Word ← "Program"`, `Code ← 'Q'` and `N ← 7`.

Expression	Value	Why
`LENGTH(Word)`	`7`	seven characters
`MID(Word, 4, 2)`	`"gr"`	two characters, starting at position 4
`LEFT(Word, 3) & "!"`	`"Pro!"`	joined with `&`
`TO_UPPER(RIGHT(Word, 2))`	`"AM"`	the inner function runs first
`ASC(Code) - ASC('A')`	`16`	`'Q'` is 81 and `'A'` is 65
`N DIV 2 + N MOD 2`	`4`	`3 + 1`
`NUM_TO_STR(N) & "th"`	`"7th"`	the number becomes a string first
`INT(N / 2)`	`3`	`3.5` cut to its whole part

Work from the inside out, and keep the quotes: `"7"` is a string and `7` is a number.

Worked example. Each statement may contain an error in its use of a function or operator. Describe the error, or write NO ERROR. (Assume every variable has the correct type.)

Statement	Error
`Result ← 2 & 4`	`&` joins strings; `2` and `4` are integers, so `+` is needed
`SubString ← MID("pseudocode", 4, 1)`	NO ERROR: one character from position 4, `"u"`
`IF x = 3 OR 4 THEN`	`OR` needs a Boolean on each side: `IF x = 3 OR x = 4 THEN`
`Result ← Status AND INT(x / 2)`	`AND` needs two Booleans; `INT(x / 2)` is an integer
`Message ← "Done" + LENGTH(MyString)`	`+` cannot add a string to an integer: `"Done" & NUM_TO_STR(LENGTH(MyString))`

Every operator works on particular types: `&` on strings, `+ - * / DIV MOD` on numbers,

`AND OR NOT` on Booleans, and `= <>` on two values of the **same** type. An "evaluate each expression, or write ERROR" table is marked the same way: `LENGTH(42)` and `"A" + 1` are ERROR, because the type does not match the function or the operator.

> **Worked example.** With `Points ← 100`, `Active ← TRUE` and `Exempt ← FALSE`, evaluate each expression.

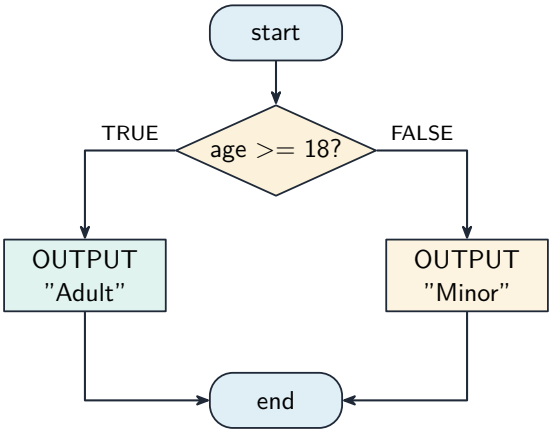
Expression	Value	Why
`(Points > 99) OR Active`	`TRUE`	both sides are true; one would do
`(Points MOD 2 = 0) OR Exempt`	`TRUE`	`100 MOD 2` is `0`
`(Points <= 75) AND (Active OR Exempt)`	`FALSE`	the first side is false, and `AND` needs both
`(Active OR NOT Active) AND NOT Exempt`	`TRUE`	`Active OR NOT Active` is always true

The last expression simplifies: `X OR NOT X` is `TRUE` whatever `X` is, so the whole expression is just `NOT Exempt`. Evaluate the brackets first, then `NOT`, then `AND`, then `OR`.

Selection

Selection chooses which steps run.

```
IF age >= 18 THEN
    OUTPUT "Adult"
ELSE
    OUTPUT "Minor"
ENDIF
```



An IF...ELSE tests the condition once, then runs exactly one branch

For more than two cases you can use a **nested** IF, but deep nesting is hard to read – a `CASE` is cleaner when testing one value against several options:

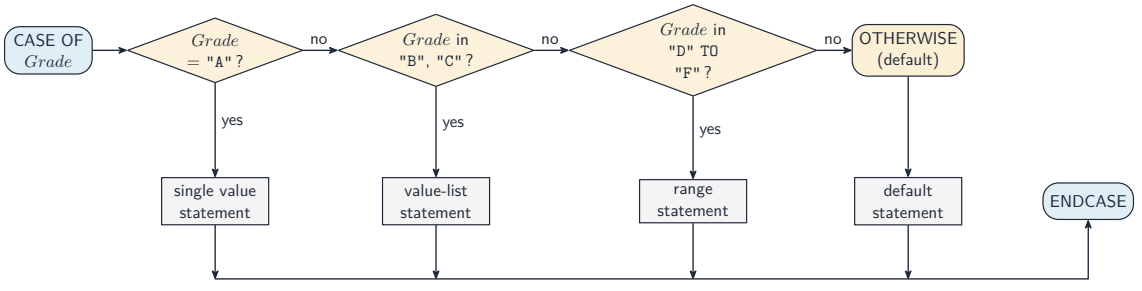
```
CASE OF Grade
    "A": OUTPUT "Excellent"
    "B": OUTPUT "Good"
    OTHERWISE: OUTPUT "Try again"
ENDCASE
```

Cambridge `CASE` allows single values, value lists (`1, 2, 3:`), and ranges (`1 TO 5:`).

A nested IF is an IF inside a branch of another IF. Each `IF` needs its own `ENDIF`, and the examiner checks that every construct is closed:

```
IF Mark >= 50 THEN
    IF Mark >= 80 THEN
        OUTPUT "Distinction"
    ELSE
        OUTPUT "Pass"
    ENDIF
ELSE
    OUTPUT "Fail"
ENDIF
```

Boundaries are where marks are lost. "A mark of 50 or more passes" is `Mark >= 50`, not `Mark > 50`; the last `CASE` branch, for "anything else", is written `OTHERWISE`, not a condition such as `> 200`. A wrong comparison here is a **logic error**: the program runs, but gives the wrong output for some inputs – and a trace table with a boundary value such as `50` is how you find it.



A CASE statement runs the branch that matches the value

Worked example. Rewrite this with the same functionality, without using a CASE structure.

```
CASE OF MySwitch
    1: ThisChar ← 'a'
    2: ThisChar ← 'y'
    3: ThisChar ← '7'
    OTHERWISE: ThisChar ← '*'
ENDCASE
```

Each value becomes a branch of a chain of IFs, and `OTHERWISE` becomes the last `ELSE`:

```
IF MySwitch = 1 THEN
    ThisChar ← 'a'
ELSE
    IF MySwitch = 2 THEN
        ThisChar ← 'y'
    ELSE
        IF MySwitch = 3 THEN
            ThisChar ← '7'
        ELSE
            ThisChar ← '*'
        ENDIF
    ENDIF
ENDIF
```

Two clauses that assign the same value are merged into one clause with a value list: `1, 2: ThisChar ← 'a'`. The guards are tested **in order**: with ranges such as `1 TO 50:` followed by `40 TO 60:`, a value of `45` takes the first branch that matches, so an assignment in a later branch may **never be performed** – and when the earlier branches already cover every possible value, the `OTHERWISE` branch is never reached either.

Going the other way, nested IFs that test several Booleans are clearer as one condition per outcome: `IF A AND B AND C THEN CALL Sub1()`, then `IF A AND B AND NOT C THEN CALL Sub2()`, and so on. Joining tests with `AND` and `OR` removes the nesting, and `IF A THEN` is accepted in place of `IF A = TRUE THEN`.

Iteration

Iteration repeats a block. Three loops differ in how many times the body runs.

Count-controlled (FOR) loop

A **count-controlled loop** – use it when you know how many times to repeat:

```
FOR i ← 1 TO 10
    OUTPUT i
NEXT i
```

A `STEP` can change the count (e.g. `FOR i ← 10 TO 1 STEP -1`). Best for a fixed number of repeats or processing each element of an **array** .

Pre-condition (WHILE) loop

A **pre-condition loop** tests the condition **before** each pass, so it may run **zero** times:

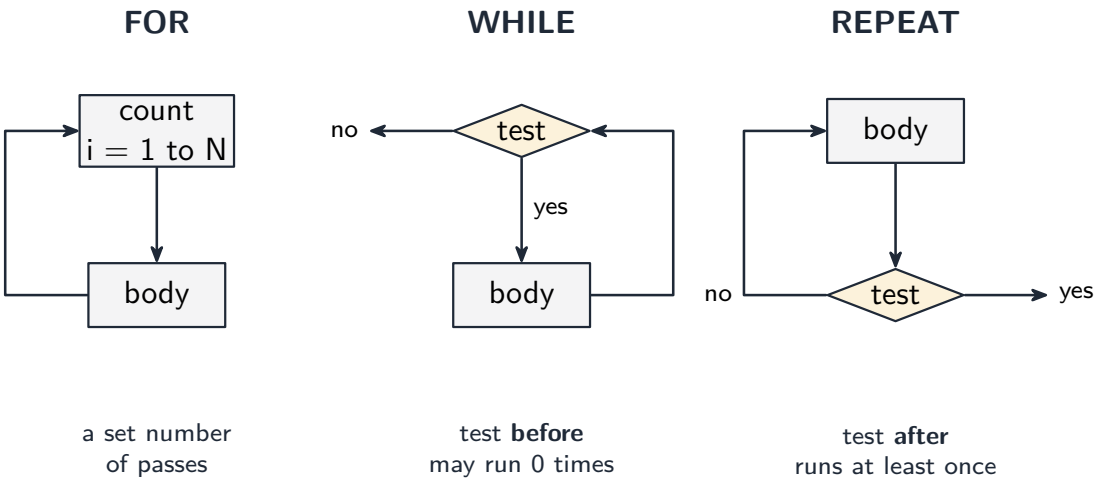
```
WHILE total < 100 DO
    INPUT n
    total ← total + n
ENDWHILE
```

Post-condition (REPEAT...UNTIL) loop

A **post-condition loop** tests the condition **after** each pass, so it always runs **at least once**:

```
REPEAT
    INPUT password
UNTIL password = correctPassword
```

Choosing the right loop



The three loops differ in where the condition is tested – before the body (WHILE), after it (REPEAT), or a set number of times (FOR)

- count known up front → **FOR**.
- may need zero passes → **WHILE**.
- always at least one pass → **REPEAT…UNTIL**.

Justify your choice by whether the count is known and whether the body must run at least once. A typical question gives a scenario ("ask for a password until correct, but always ask at least once") and asks which loop fits.

The two marks are for the **name** of the loop and the **reason**, in the scheme's words: *count-controlled, because the number of iterations is known before the loop starts*; *post-condition, because the loop body must be executed at least once*; *pre-condition, because the loop may not need to execute at all*. A loop over the four elements of an array that has been written as a `WHILE` with a counter is "not the most appropriate": the count, four, is known, so a `FOR` loop fits.

> **Worked example.** Which loop suits each task? (a) print the 12 times table; (b) keep reading numbers until the user enters 0; (c) ask for a password until it is correct. Choose by asking **how many times** the body runs and **when the test happens**. (a) The count is known in advance (12), so use a **FOR** loop. (b) The count is unknown, and the very first input might already be 0 - so the test must come **before** the body: a **WHILE** loop, which runs **zero** or more times. (c) The count is unknown, but you must always ask **at least once** before there is anything to test - so the test comes **after** the body: a **REPEAT…UNTIL**, which runs **one** or more times. The deciding question is whether the body must run at least once: WHILE may run zero times, REPEAT always runs once.

■ Dry running with a trace table

A **trace table** records the value of each variable as you **dry run** (work through by hand) an algorithm. It is how you test a loop on paper, and a six-mark question on most Paper 2s.

```
DECLARE Count, Total : INTEGER
Count ← 1
Total ← 0
WHILE Total < 10
    Total ← Total + Count * 2
    Count ← Count + 1
ENDWHILE
OUTPUT Count, Total
```

Count	Total	Total < 10	OUTPUT
1	0	TRUE	
2	2	TRUE	
3	6	TRUE	
4	12	FALSE	4, 12

Rules that earn the marks: one column per variable, in the order the question gives; write a value only when it **changes**; start a new row each time the loop repeats; evaluate the condition with the current values, and stop the moment it is `FALSE`; put the output in its own column, exactly as it would appear. Trace the algorithm as written, not the one you think was intended – if it never stops, say so.

> **Worked example.** Which constructs does each line use – selection, iteration or a subroutine call?

Pseudocode	Selection	Iteration	Subroutine
IF Ready = TRUE THEN			

```
CALL Start()
```

ENDIF | `FOR I ← 1 TO 20 ... NEXT I` | | yes | | | `WHILE NOT IsFull() ... ENDWHILE` | | yes | yes | | `CASE OF Key ... OTHERWISE ... ENDCASE` | yes | | |

`IF` and `CASE` are selection; `FOR`, `WHILE` and `REPEAT` are iteration; a name followed by brackets – `Start()`, `IsFull()` – is a call to a procedure or a function, wherever it appears, including inside a condition.

Procedures and functions

Structured programming builds a program from small named **subroutines**, each with one job.

Procedure

A **procedure** is a named block that **does an action**; it may take **parameters** but does **not return a value**.

```
PROCEDURE Greet(name : STRING)
    OUTPUT "Hello, ", name
ENDPROCEDURE

CALL Greet("Ada")
```

Function

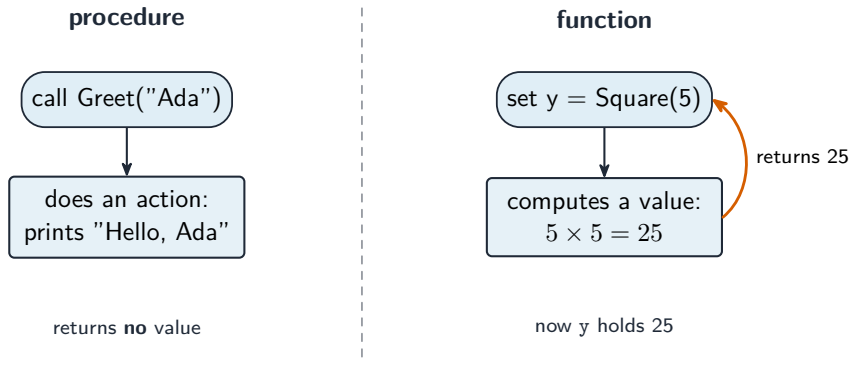
A **function** is like a procedure but it **returns a value** that becomes part of an expression.

```
FUNCTION Square(x : INTEGER) RETURNS INTEGER
    RETURN x * x
ENDFUNCTION

result ← Square(5) + 1     // result = 26
```

Use a **procedure** when the subroutine performs an action; use a **function** when it computes a value for the caller.

The syllabus asks *where in the construction of an algorithm* each is appropriate. A procedure is appropriate where the same group of steps is needed at several points (validate an input, print a menu, swap two values): the steps are written once and `CALL`ed by name. A function is appropriate where a single value must be calculated and then used in an expression – a total, a `TRUE`/`FALSE` result, the larger of two numbers – because the **return value** replaces the call: `IF IsValid(Code) THEN`.

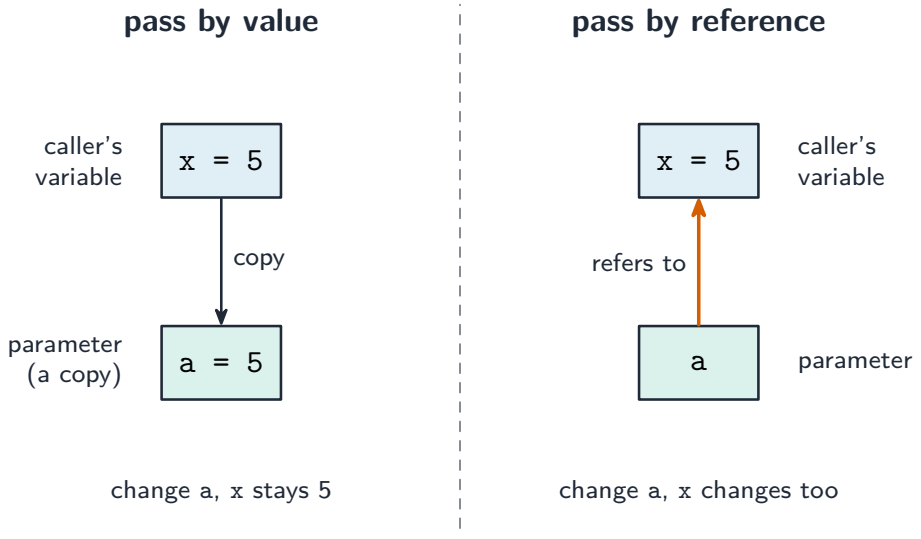


A procedure does an action and returns nothing; a function returns a value you use in an expression

Parameters

A **parameter** is a variable a subroutine declares to receive input; the values the caller supplies are **arguments** . Two ways to pass them:

- **pass by value** – the routine gets a **copy**; changes inside it do not affect the caller. Use for inputs it only reads.
- **pass by reference** – the routine gets a **reference** to the caller's variable; changes **do** affect the caller. Use when it must update a parameter.



Pass by value copies the value into a new box; pass by reference lets the routine change the caller's own variable

```
PROCEDURE Swap(BYREF a : INTEGER, BYREF b : INTEGER)
    DECLARE temp : INTEGER
    temp ← a
    a ← b
    b ← temp
ENDPROCEDURE
```

Cambridge pseudocode writes the mode in the header, `BYVAL` or `BYREF`, before each parameter. If neither is written, `BYVAL` is assumed, so a routine that must change the caller's variable – `Swap`, or a procedure that updates a running total – needs `BYREF` in its header.

Worked example. What is output?

```
PROCEDURE Adjust(BYREF X : INTEGER, BYVAL Y : INTEGER)
    X ← X + Y
    Y ← Y * 2
ENDPROCEDURE

A ← 5
B ← 3
CALL Adjust(A, B)
OUTPUT A, B
```

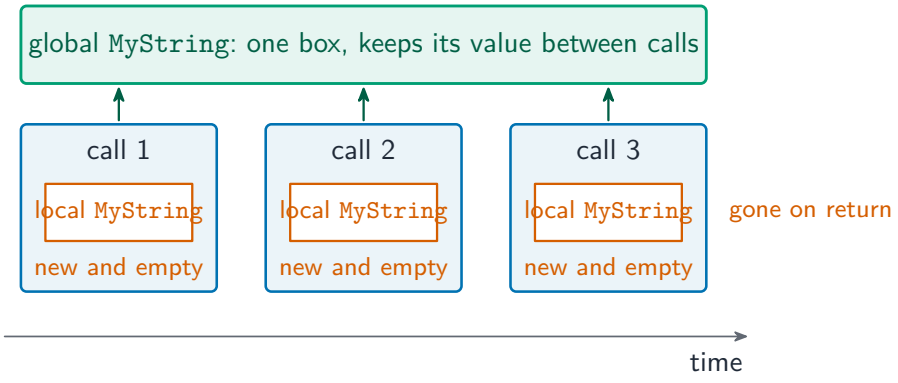
`X` is a reference to `A`, so `A` becomes `8`. `Y` is a copy of `B`, so doubling `Y` leaves `B` at `3`. The output is `8, 3`. Had the header said `BYVAL X`, `A` would still be `5`.

■ Local vs global variables

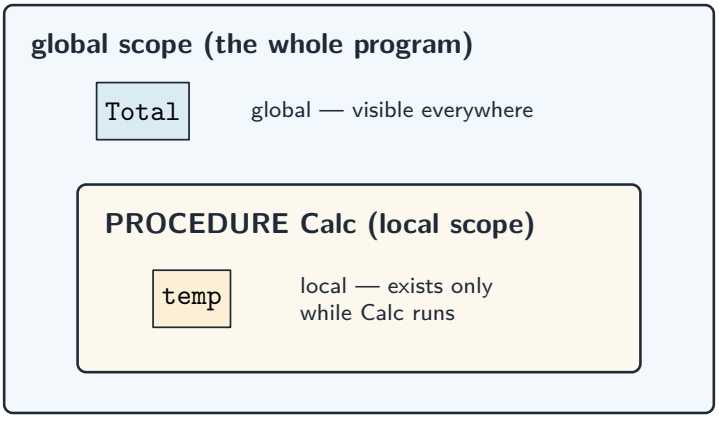
A **local variable** is declared inside a subroutine and exists only while it runs. A **global variable** is declared outside and is visible everywhere. Prefer locals and parameters – heavy use of globals makes code hard to follow and test. (The region where a name is visible is its **scope** .)

The one-line difference: a global variable can be accessed from anywhere in the program, a local variable only inside the subroutine that declares it. Benefits of local variables the scheme accepts: the same identifier can be used in another subroutine without a clash; the value cannot be changed accidentally by other parts of the program; the memory is released when the subroutine ends; and the subroutine is **self-contained**, so it can be tested on its own and reused in another program.

A local variable is created **each time** the subroutine is called and destroyed when it returns, so it cannot carry a value from one call to the next. A procedure that builds up a string over repeated calls therefore needs that string to be global (or passed `BYREF`). If `MyString` is changed from a global to a local declared inside `MyOutput()`, every call starts with a new, empty `MyString`, the text added by earlier calls is lost, and the procedure "does not work as expected".



A local variable is a new, empty box on every call; only a global variable (or a BYREF parameter) keeps a value between calls



A global variable is visible everywhere; a local variable exists only inside its own procedure

■ When to use a subroutine

Use a subroutine when:

- the same logic appears in **more than one place** – write it once, call it many times.
- a block has a **clear named purpose** – the name documents what it does.
- the program is **complex** – break it into parts (**decomposition**).
- you want to **test a piece in isolation**.

Don't make them so tiny that the call costs more than the work inside.

Terminology

- **definition** – the `PROCEDURE ... ENDPROCEDURE` (or function) block.
- **call** – where it is invoked. **argument** – a value passed in. **parameter** – the variable that receives it.
- **return value** – what a function passes back.
- **procedure/function header** – the first line giving the name and parameters (`PROCEDURE Name(params)` or `FUNCTION Name(params) RETURNS type`).
- **procedure/function interface / signature** – name + parameters + return type: what a caller must know to use it.

> **Worked example.** Describe each term used in the header `FUNCTION Pass2(Count : INTEGER) RETURNS BOOLEAN`.

Term	Meaning
`FUNCTION`	a subroutine that returns a value
`Pass2`	the identifier used to call it
`Count`	the parameter: the identifier that receives the argument passed in
`INTEGER`	the data type of the parameter
`RETURNS BOOLEAN`	the data type of the value the function returns

The two identifiers in `PROCEDURE MyProc(Count : INTEGER, Message : STRING)` are **parameters**: they receive the values passed in when the procedure is called, and are used inside it like local variables.

To convert a procedure into a function: change `PROCEDURE` to `FUNCTION` and add `RETURNS <type>`; replace the `OUTPUT` (or the `BYREF` parameter that carried the result out) with a `RETURN` statement; and change every call so that the returned value is used, `Result ← Unpack(Text)` instead of `CALL Unpack(Text, Result)`. For a "write the header" question, write the whole line: `FUNCTION Calculate(Expression : STRING) RETURNS INTEGER`. An array parameter is passed **by reference**, so a procedure that writes into an array changes the caller's array.

When a program gains a new module, the interface is what is agreed first: the name, the parameters (how many, in what order, of what type) and the return type, plus any global data the module reads or writes. A module that sends a reminder before a due date needs the record (or its index) as a parameter and returns nothing, so it is a procedure; the main program calls it once per record.

Writing a module for Paper 2

Half of Paper 2 is "write pseudocode for module X". The scheme awards a mark per feature, so a module that is not finished still scores for every correct part. The parts the examiner looks for:

```
FUNCTION CountAbove(BYVAL Limit : INTEGER) RETURNS INTEGER
    DECLARE Index, Count : INTEGER
    Count ← 0
    FOR Index ← 1 TO 50
        IF Score[Index] > Limit THEN
            Count ← Count + 1
        ENDIF
    NEXT Index
    RETURN Count
ENDFUNCTION
```

header: name, parameters, return type
local variables declared with a type
total initialised before the loop
loop over every element
the condition, with the right boundary
the update, inside the IF
every construct closed
the return value, after the loop

Each part of a module answer carries its own mark, so write all of them even when one is uncertain

1. **The header**, as the question describes it: `PROCEDURE Name(Param : TYPE)` or `FUNCTION Name(Param : TYPE) RETURNS TYPE`, with `BYREF` where the routine must change the argument.
2. **Local declarations**: `DECLARE` every local variable with its type, and initialise counters and totals (`Count ← 0`).
3. **The loop** that visits every element: `FOR Index ← 1 TO 50` for an array whose size is given; `WHILE NOT EOF(...)` for a file.
4. **The condition**, with the right comparison and boundary, on the right item: `IF Score[Index] > Limit THEN`.
5. **The update** inside the branch: the count increased, the value stored, or the message output.
6. **The end**: `RETURN` once, after the loop, in a function; `ENDFUNCTION` or `ENDPROCEDURE`; and every `IF`, `FOR` and `WHILE` closed.

Worked example. A global array `Score : ARRAY[1:50] OF INTEGER` holds test scores. Write a function `CountAbove(Limit : INTEGER)` that returns how many scores are greater than `Limit`.

```
FUNCTION CountAbove(BYVAL Limit : INTEGER) RETURNS INTEGER
    DECLARE Index, Count : INTEGER
    Count ← 0
    FOR Index ← 1 TO 50
        IF Score[Index] > Limit THEN
            Count ← Count + 1
        ENDIF
    NEXT Index
    RETURN Count
ENDFUNCTION
```

Marks: the header with its parameter and `RETURNS INTEGER`; `Count` declared and set to `0`; a loop over all 50 elements; the comparison `> Limit` (not `>=`); the count updated inside the `IF`; `RETURN Count` after the loop. The main program uses the return value in an expression or an output: `OUTPUT "Above 70: ", CountAbove(70)`.

Worked example. Write a function `IsValid(Code : STRING)` that returns `TRUE` when `Code` is two capital letters followed by four digits – the **format** `AB1234` – and `FALSE` otherwise.

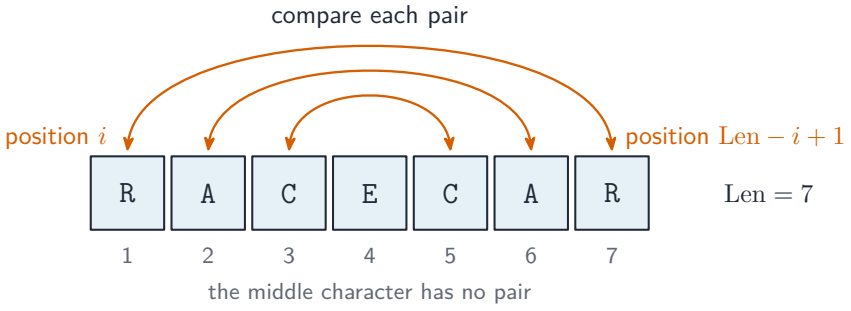
```
FUNCTION IsValid(BYVAL Code : STRING) RETURNS BOOLEAN
    DECLARE Index : INTEGER
    DECLARE Ch : STRING
    IF LENGTH(Code) <> 6 THEN
        RETURN FALSE
    ENDIF
    FOR Index ← 1 TO 6
        Ch ← MID(Code, Index, 1)
        IF Index <= 2 THEN
            IF Ch < "A" OR Ch > "Z" THEN
                RETURN FALSE
            ENDIF
        ELSE
            IF Ch < "0" OR Ch > "9" THEN
                RETURN FALSE
            ENDIF
        ENDIF
    NEXT Index
    RETURN TRUE
ENDFUNCTION
```

The length check comes first, so `MID` is never asked for a position that does not

exist. **Validation** like this returns a `BOOLEAN` so the caller can write `IF IsValid(Entry) THEN ... ELSE OUTPUT "Invalid code" ENDIF`: a message to the user is output by the caller, not by the function – a function calculates, a procedure acts.

> **Worked example.** Write a function `IsPalindrome(Word : STRING)` that returns `TRUE` when `Word` reads the same backwards, such as `"RACECAR"`.

Compare the characters from the two ends, moving inwards: position `Index` is paired with position `Len - Index + 1`, and only the first half needs testing.



A palindrome check pairs position `i` with position `Len - i + 1` and stops at the middle

```
FUNCTION IsPalindrome(BYVAL Word : STRING) RETURNS BOOLEAN
    DECLARE Len, Index : INTEGER
    Len ← LENGTH(Word)
    FOR Index ← 1 TO Len DIV 2
        IF MID(Word, Index, 1) <> MID(Word, Len - Index + 1, 1) THEN
            RETURN FALSE
        ENDIF
    NEXT Index
    RETURN TRUE
ENDFUNCTION
```

The same three tools – a `FOR` over the positions, `MID(s, i, 1)` to read one character, and `&` to build a new string – answer most string modules on Paper 2: counting how often a character occurs (`IF MID(s, i, 1) = Ch THEN Count ← Count + 1`), replacing every instance of a character (add either `NewChar` or the original character to `NewString` at each position), hiding all but the last four digits of a card number (add `'*'` for every position up to `Len - 4`), or writing your own `MID()` by joining the characters from `Start` to `Start + Length - 1`. Asking `MID` for a position past the end of the string is a **run-time error**, so check `LENGTH` first.

Files. Values in variables disappear when the program ends, so a module that must keep data for the next run writes it to a file: `OPENFILE "scores.txt" FOR WRITE`, one

`WRITEFILE "scores.txt", NUM_TO_STR(Score[Index])` per line inside the loop, and `CLOSEFILE "scores.txt"` once, after the loop; reading back uses `FOR READ`, `READFILE` and `WHILE NOT EOF("scores.txt")`. Topic 10 has the full file section; here the marks are for opening in the right mode, the read or write inside the loop, and closing once after it.

Writing efficient pseudocode

Three features that make pseudocode easier to understand – the answer to a "state three features" question – are **meaningful identifiers** (`Total`, not `t`), **indentation** of the statements inside each construct, and **comments** (`// ...`) that explain the purpose; keywords in capitals, one statement per line and blank lines between sections are also accepted. Efficient pseudocode goes further:

- **move invariants out of loops** – if a value (an **invariant**) does not change with the loop counter, compute it once before the loop.
- **exit a loop early** when the answer is found (stop a **linear search** as soon as the target appears).
- **avoid redundant work** – store a result and reuse it instead of recomputing.
- **choose the right data structure** – an array beats many separate variables when the items belong together.
- **replace deep nested IFs with CASE** when testing one value against many.
- **comment the intent**, not the mechanics (`// validate the postcode`, not `// loop 6 times`).
- **use meaningful names** (`numberOfPupils`, not `n`) and **initialise variables** before use.

slower

```
FOR i = 1 TO n
 x = slow() // every pass
 use(x, i)
NEXT i
```

faster

```
x = slow() // once
FOR i = 1 TO n
 use(x, i)
NEXT i
```

Move unchanging work out of the loop so it runs once

Testing and errors

Three kinds of error, each found in a different way:

Error	What it is	Example	Found by
syntax error	a statement that breaks the rules of the language	a missing `ENDIF`; `OUTPT "Hi"`	the translator, before the program runs
run-time error	the program runs, but a statement cannot be carried out	division by zero; an array index of 0 or 51; a function called with an invalid parameter; a loop that never ends, so the program "freezes"	while running: the program stops or hangs
logic error	the program runs to the end, but the output is wrong	`>` where `>=` was needed; a total never set to `0`	testing with a trace table and chosen test data

An **IDE** helps find the last two: a **breakpoint** stops the program at a chosen line; **single stepping** then runs one statement at a time; and the report (or watch) window shows the value of each variable at that moment, so the line where a value goes wrong is seen directly. Test methods and test data are in topic 12.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly.

Term	Definition
procedure	a subroutine that carries out a task (a sequence of steps) and does not return a value; it is called with `CALL`
function	a subroutine that returns a single value to the point where it was called, so it can be used in an expression
parameter	the identifier in a subroutine header that receives a value or a reference when the subroutine is called
argument	the value (or variable) supplied in the call, matched to a parameter
passing by value	a copy of the argument's value is given to the subroutine, so changes inside it do not affect the original variable

Term	Definition
passing by reference	the address of the variable is given to the subroutine, so changes inside it change the original variable
header	the first line of a subroutine definition: its name, its parameters and, for a function, its return type
interface	what a calling program must know to use a subroutine: its name, its parameters (number, order, type) and its return type
return value	the value a function passes back to the expression that called it
local variable	declared inside a subroutine; it exists only while the subroutine runs and can be used only inside it
global variable	declared outside every subroutine; it can be used anywhere in the program
count-controlled loop	repeats a fixed number of times, controlled by a counter (`FOR ... NEXT`)
pre-condition loop	tests its condition before each iteration, so the body may never run (`WHILE ... ENDWHILE`)
post-condition loop	tests its condition after each iteration, so the body runs at least once (`REPEAT ... UNTIL`)
constant	a named value that cannot change while the program runs
subroutine	a self-contained block of code that performs a task and is called by name: a procedure or a function
library routine	a subroutine that has already been written and tested, and is available to be called from a program

Exam tips

- Distinguish a **procedure** (no return value) from a **function** (returns a value); know **pass by value vs by reference**.
- Choose the right loop: **count-controlled (FOR)** when the number of repeats is known, **condition-controlled (WHILE/REPEAT)** otherwise.
- Distinguish **local vs global** variables and scope; prefer local variables in reusable modules.

- Use the insert's exact routine names and parameter order. `VAL` and `STR` are IGCSE names and score nothing; `UCASE` and `LCASE` are real 9618 routines from the Pseudocode Guide but act on **one character**, so on Paper 2 a whole string takes `TO_UPPER` or `TO_LOWER`.
- In a "write pseudocode" answer the header, the declarations, the loop, the condition, the update and the `RETURN` each carry a mark: write all six parts, even if one is uncertain.

■ Common mistakes

- Calling a function and not using what it returns. Assign the result, or use it in the expression or output: `Sorted ← BubbleSort(MyArray, 7)`.
- Passing a length one out: `6` for a seven-element array, or the last index where the length was wanted. Decide whether the parameter is a length or an index, and check that the last element is visited.
- Closing a file inside the loop that reads it. Open once, close once, after the loop.
- Using the input as a filename directly. Add the extension the question gave: `File Name ← Choice & ".txt"`.
- Leaving constructs open. Every `IF` needs its `ENDIF`, every `FOR` its `NEXT`, every `WHILE` its `ENDWHILE`, and every function its `RETURN`; the scheme has a mark for it.
- Wrong boundaries: `>` for "at least" (which is `>=`), or a `FOR` that starts at `0` for an array declared `[1:50]`.
- A counter or total that is never set to `0` before the loop.
- In a trace table, rewriting every variable on every row, or changing a value before the statement that changes it has run.
- Half a condition: `IF x = 3 OR 4` – each side of `OR` and `AND` must be a complete comparison. And `+` does not join strings; `&` does.
- Declaring as local a value that must survive between calls. A running total or a string built up over several calls is global or `BYREF`.

Software Development

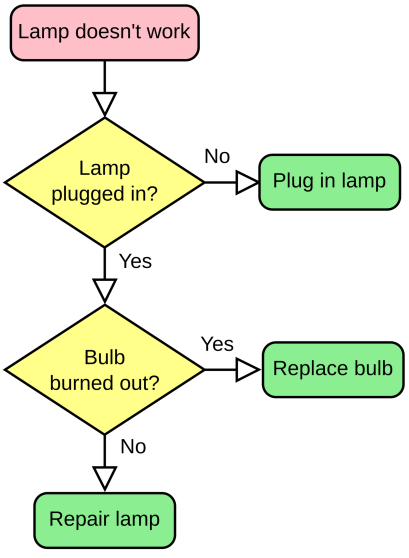
A-Level Computer Science

Program development life cycle

A **development life cycle** is the set of stages from idea to finished, maintained software. It exists to **plan, manage and control** a project – to build the right product, on time, with good quality.



Software is built by teams who follow a development life cycle to stay coordinated



A flowchart plans a program's logic during the design stage of the cycle

Why a life cycle is needed

The examiner's list for "the purpose of a development life cycle": it breaks a large project into **stages** that can be planned and managed; it makes sure the **requirements** are found and agreed before design and coding begin; it builds in **testing** and **documentation** rather than leaving them to the end; it lets the team track progress against **milestones** and manage risk; and it gives the customer defined points at which

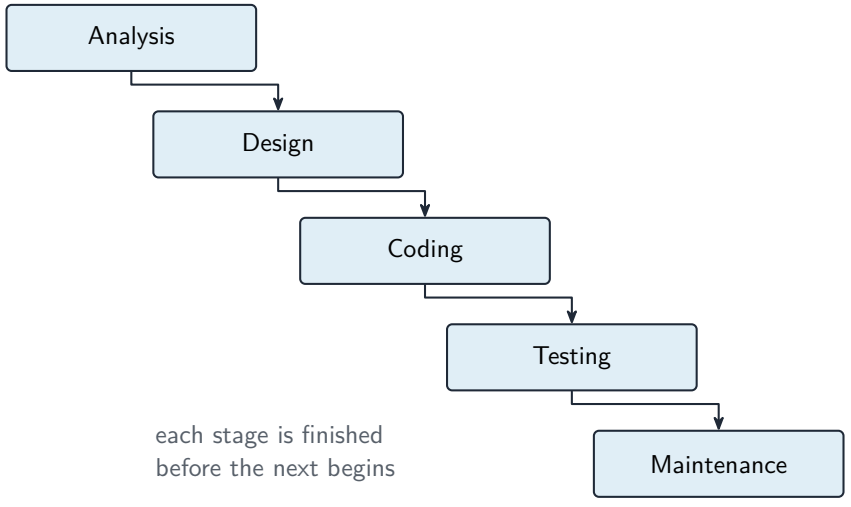
to review the work. Without one, a team codes first and discovers late that it built the wrong thing.

Why there are different ones

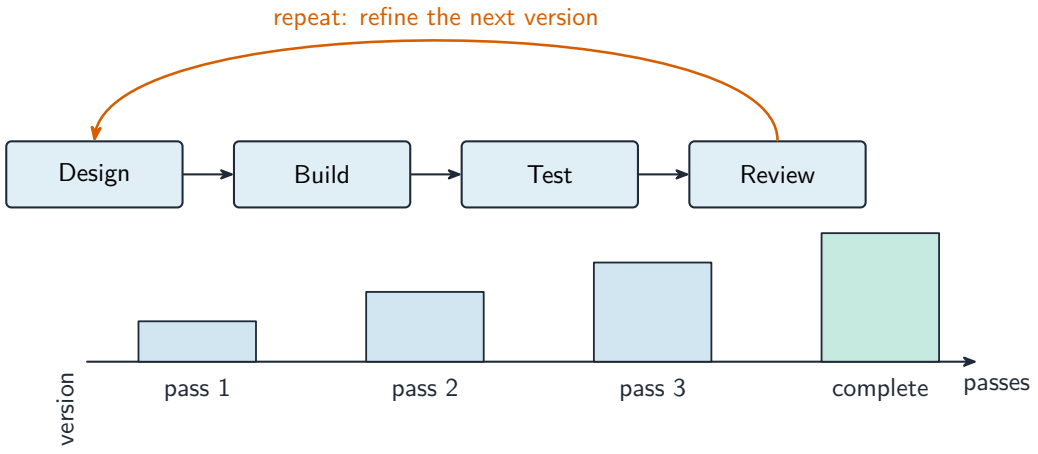
No single life cycle fits every project, so several **development life cycles** exist. The choice depends on the size and complexity, how clear the **requirements** are at the start, how much change is expected, the risk level, the team, and the deadline.

Common models

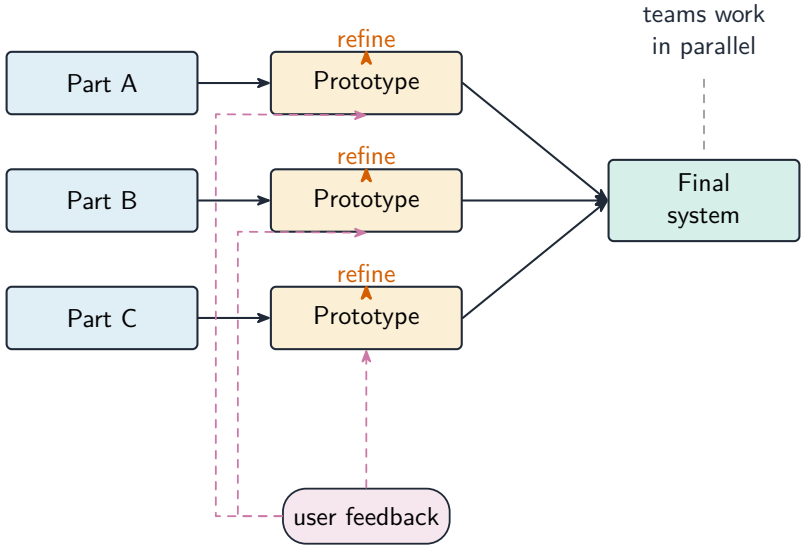
- **Waterfall** – a linear sequence (Analysis → Design → Coding → Testing → Maintenance), each stage finished before the next. Clear and well-documented; good for **stable requirements**, but poor at coping with mid-project change, and the customer sees nothing working until the end.
- **Iterative model** – repeated passes, each producing a partial version that is reviewed and refined. Catches problems earlier; good when requirements are **discovered over time**, but harder to estimate.
- **Rapid Application Development** (RAD) – heavy use of a **prototype** and user feedback. Very fast first delivery; good for **changing requirements**, but depends on user availability and suits smaller systems.
- **Agile** – short iterations ("sprints"), constant collaboration and testing. Flexible and adaptive, but needs a committed customer and a skilled team.



The waterfall model: each stage is finished before the next begins



The iterative model: repeated passes refine the program



Rapid application development: teams work on parts in parallel

Principles, benefits and drawbacks – as the mark scheme lists them.

Model	Principle	Benefits	Drawbacks
waterfall	the stages run in a fixed order, each completed and signed off before the next starts; going back means restarting the sequence	simple to manage; every stage is fully documented; requirements are fixed early, so costs and dates can be estimated	inflexible once a stage is finished; no working software until late; a mistake in analysis is expensive to fix later; the customer cannot see progress
iterative	a small working version is built first, then repeatedly improved through further versions until complete	working software early and often; problems found in early versions; the customer's feedback shapes each version; requirements can change	hard to estimate the total time and cost; repeated testing costs effort; needs the customer to be available; can drift if versions are not planned
RAD	prototypes of parts of the system are built quickly and refined with the user until accepted, often in parallel by several teams	very fast delivery of a first version; the user is involved throughout, so the product fits their needs; changes are easy to absorb	needs skilled developers and committed users; documentation is weak; less suited to large or safety-critical systems

Worked example. A company must be the first to launch a website for a new games console, and the design will change as the console's features are announced. Name the most suitable life cycle and justify it.

RAD. A prototype of the site can be built and shown to the users within days, and refined as the requirements change; the site is small enough for a prototype-driven approach, and speed of delivery is the main requirement. Waterfall would fix the requirements before any page was built and deliver nothing until the end.

■ The standard stages

Each stage has a purpose, an output and typical activities – a "describe the … stage" question wants two or three of these.

- **analysis** – find out **what** the program must do. Activities: interviews, questionnaires and observation of the current system; a feasibility study; agreeing the **requirements** specification, which every later stage is checked against.
- **design** – decide **how** it will do it. Outputs: the **structure chart** (modules and parameters), **flowcharts** or **pseudocode** for each module, **identifier tables** and

data structures, screen and file layouts, and the **test plan** written *now*, from the specification, before any code exists.

- **coding** (**implementation**) – write the program in a high-level language, module by module, following the design; each module is tested as it is written.
- **testing** – run the program against the test plan (normal, abnormal, extreme and boundary data) and correct the errors found; integration, alpha, beta and acceptance testing follow.
- **maintenance** – after release, correct faults, adapt the program to new hardware, software or law, and improve it (see below).

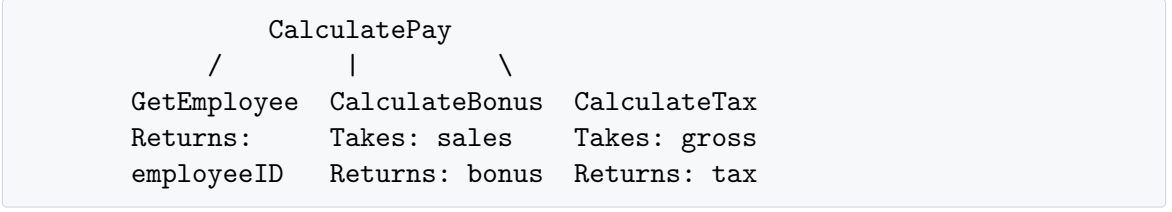
> **Worked example.** Complete the waterfall diagram *Analysis* → *?* → *?* → *?* → *Maintenance* and describe what happens at the design stage.

The missing stages are **Design, Coding, Testing**. At the design stage the requirements are turned into a plan for the program: the problem is decomposed into modules (a structure chart), the algorithm for each module is written as pseudocode or a flowchart, the data structures and identifiers are chosen, the screens and files are laid out, and the test plan is written from the specification.

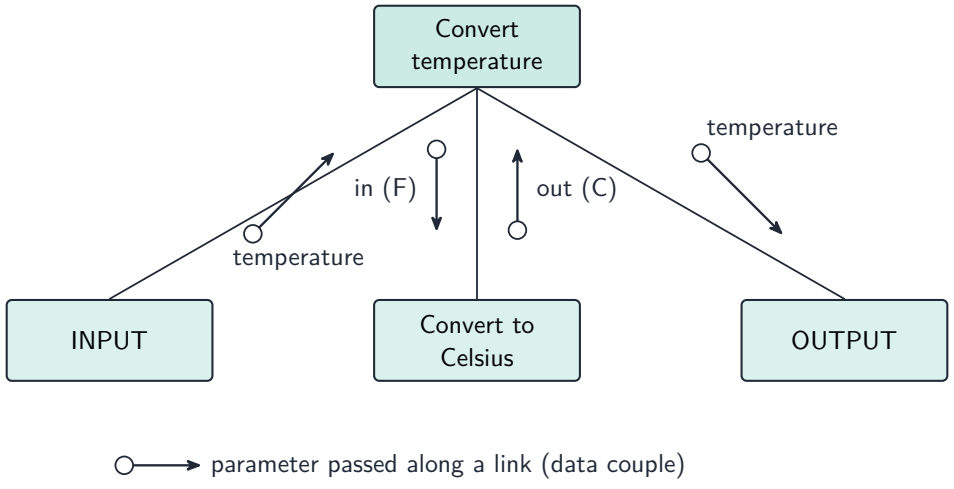
Program design tools

■ Structure chart

A **structure chart** shows the **hierarchical decomposition** of a program into modules (**subroutines**) and the **parameters** passed between them. Each module is a rectangle; lines link caller (above) to callee (below); small arrows show data going down and results coming back up. The design can then be turned into equivalent **pseudocode** .

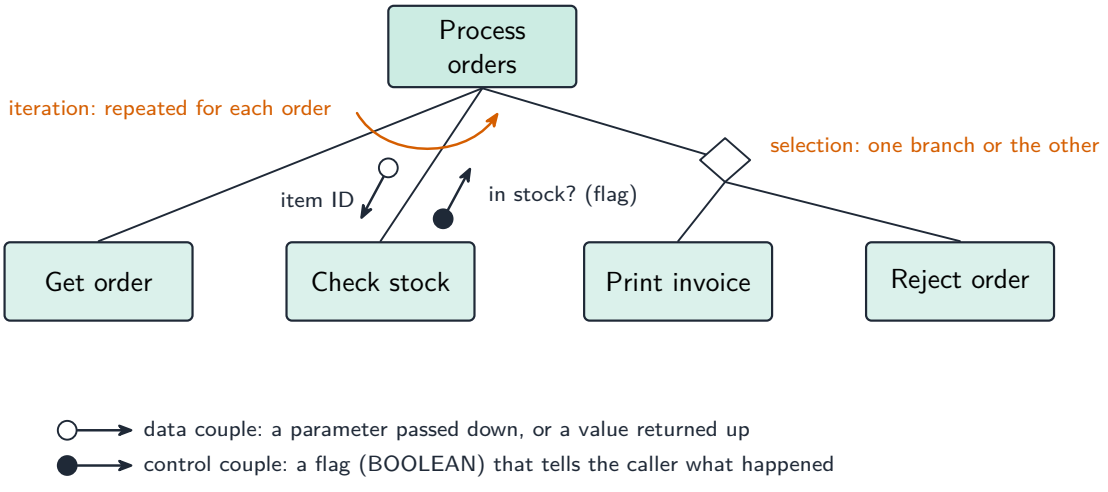


It is a design-stage tool, and you can read the procedure signatures off it.



A structure chart: modules with the parameters passed between them

The symbols the examiner asks about. A box is a module; a line links a caller (above) to the modules it calls (below), read left to right in the order they are called. A small arrow with an **open circle** at its tail is a **data couple** – a parameter passed *down* into a module or a value returned *up*; an arrow with a **filled circle** is a **control couple**, a flag (usually BOOLEAN) that tells the caller what happened. A **diamond** at a branch means **selection**: only one of the modules below it is called, depending on a condition. A **curved arrow** sweeping across the links means **iteration**: the modules under it are called repeatedly in a loop.



The structure-chart symbols: data and control couples, a selection diamond and an iteration arrow

> **Worked example.** Four modules are defined as `PROCEDURE Main()`, `PROCEDURE ReadData(BYREF Count : INTEGER)`, `FUNCTION IsValid(Value : INTEGER) RETURNS BOOLEAN` and `PROCEDURE Report(Total : INTEGER, Count : INTEGER)`. Main calls ReadData, then calls IsValid once for each value read, then calls Report. Describe the structure chart.

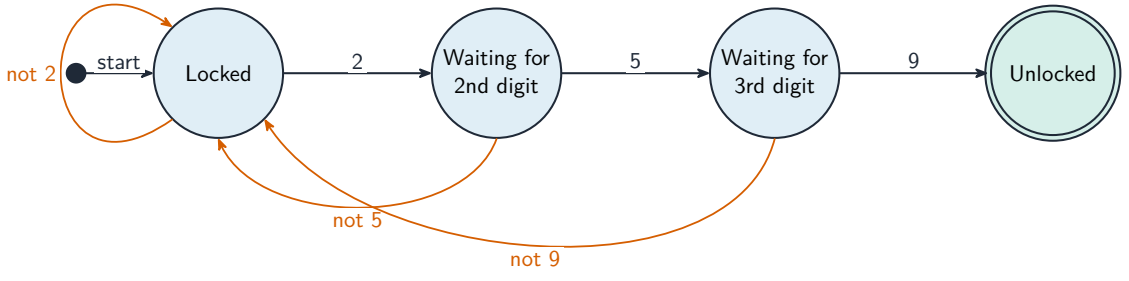
Main at the top; ReadData, IsValid and Report in a row beneath it, left to right in calling order. On the ReadData link an upward data couple `Count` (a BYREF parameter comes back). On the IsValid link a downward data couple `Value` and an upward control couple (the BOOLEAN result), with a curved iteration arrow across that link because it is called for each value. On the Report link two downward data couples, `Total` and `Count`. Reading the other way, a function is any module that returns a value – its header needs `RETURNS` and the returned type.

State-transition diagram

A **state-transition diagram** shows the **states** a system can be in and the **events** that move it between them – good for vending machines, traffic lights, user interfaces. **State-transition diagrams** are used to document the behaviour of an algorithm or system. Each state is a circle; each transition is an arrow labelled with the event.

```
   coin inserted                item selected
[Idle] ---------------→ [Awaiting selection] -----------→ [Dispensing]
```

It makes missing transitions easy to spot ("what if a second coin is inserted while awaiting selection?").



A state-transition diagram for a door lock with code 259

Reading and drawing one. Each transition is labelled *input / output* (or *condition / action*): what happened, then what the system does as it changes state. A question gives a table of *current state, input, output, next state* and asks for the diagram, or the reverse – every row of the table is exactly one arrow. Check that every state has an arrow leaving it for every input that can occur, including the ones that leave the

state unchanged (an arrow that loops back to the same state).

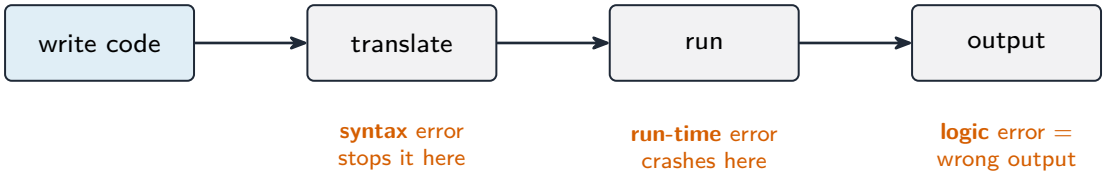
> **Worked example.** A pump controller has states *pump off* and *pump on*. In *pump off*, the input *low level detected* produces the output *activate pump* and moves to *pump on*; in *pump on*, *normal level detected* produces *deactivate pump* and moves to *pump off*. Any other input leaves the state unchanged. Draw the table.

Current state	Input	Output	Next state
pump off	low level detected	activate pump	pump on
pump off	normal level detected	–	pump off
pump on	normal level detected	deactivate pump	pump off
pump on	low level detected	–	pump on

The two "no change" rows become loop arrows on the diagram; leaving them out loses the mark for completeness.

Errors

- **syntax error** – breaks the language's grammar (missing bracket, misspelled keyword). Caught at translation time; the program won't run until fixed.
- **run-time error** – happens while running (divide by zero, file not found, array index out of range). The program crashes or raises an exception; fix by adding checks.
- **logic error** – the program runs but gives **wrong results** (using `+` for `-`, an off-by-one loop, conditions in the wrong order). The hardest to find; the only sign is wrong output, so use careful testing and tracing.



When each error shows up: syntax at translation, run-time during the run, logic in the output

Exposing and avoiding faults. Faults are *exposed* by testing against a test plan, by a dry run or trace table, by a walkthrough with colleagues, and by the IDE's debugger (breakpoints, single stepping, watching variables). They are *avoided* by designing before coding (structure chart, pseudocode), by modular code with meaningful identifiers and comments, by **validation** of every input, by handling exceptions rather than letting a run-time error crash the program, and by the IDE's dynamic syntax checks as you type.

Worked example. State the type of error in each case and how it shows itself. (a) `Result <- STR_TO_NUM(x) / STR_TO_NUM(y)` is run with `y = "0"`. (b) The same line is run with `x = "12a"`. (c) A loop written as `FOR i <- 1 TO 9` processes a ten-element array. (d) `OUTPUT "Total: " Total` is missing a comma.

(a) **Run-time error** – division by zero; the program crashes when this line is executed with that data. (b) **Run-time error** – the string cannot be converted to a number. (c) **Logic error** – the program runs but the tenth element is never processed, so the output is wrong. (d) **Syntax error** – the statement breaks the language's rules and is reported by the translator before the program runs.

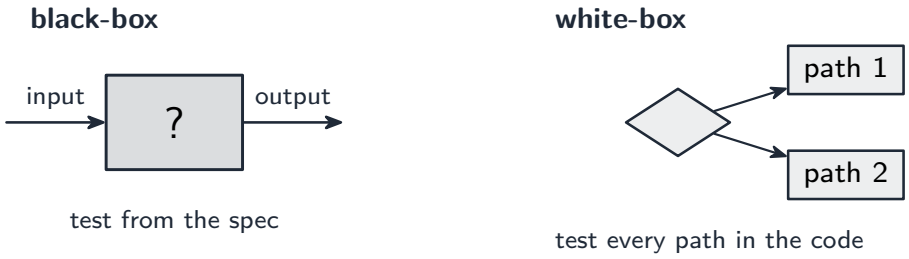
Worked example. Correct the errors in this pseudocode, which should output the average of ten marks.

```
Total <- 0
FOR i <- 1 TO 10
    INPUT Mark
    Total <- Total + Mark
NEXT i
Average <- Total / 9
OUTPUT "Average" Average
```

The division should be by 10, not 9 (a logic error); the output line needs a comma or an `&` between the string and the value (a syntax error); and `Average` is never declared as REAL (a syntax or run-time error, depending on the language). Say which line and what the corrected line is: `Average <- Total / 10`.

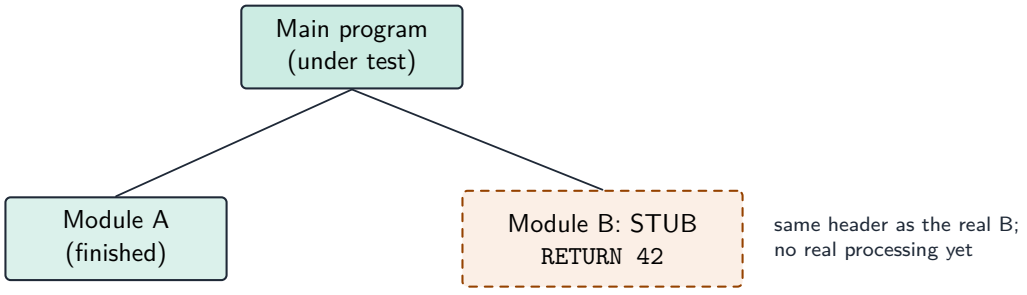
Testing methods

- **dry run** – trace the code on paper, writing each variable's value in a table.
- **walkthrough** – a team review of the code.
- **white-box testing** – designed from the code's internal structure, covering every statement, branch and loop.
- **black-box testing** – designed from the specification only: feed inputs, check outputs.
- **integration testing** – combine modules and test the interfaces between them.
- **alpha testing** α – by the developers/in-house before release; **beta testing** β – by a limited group of real users in their own environment.
- **acceptance testing** – by the customer, to decide if the product is fit for purpose.
- **stub** – a placeholder for a module that does not exist yet, so the structure can be tested top-down.



Black-box tests the specification; white-box tests the code paths

Which method, when. A **dry run** and a **walkthrough** need no computer – the dry run is you, tracing the algorithm with a **trace table** ; the walkthrough is a meeting in which the author explains the code line by line and colleagues look for faults, so it also spreads knowledge of the code through the team and checks it against the design. **White-box** tests are written by someone who can see the code and aims to exercise every path; **black-box** tests are written from the specification and check only inputs against expected outputs, so a user or a separate tester can do them. **Integration** testing follows module testing: modules that pass alone can still fail when the data passed between them is the wrong type or in the wrong order. **Alpha** testing is in-house; **beta** testing gives a release candidate to a sample of real users, who report faults from real use; **acceptance** testing is the customer checking the finished product against the requirements before paying for it. A **stub** lets top-down testing start before every module exists.



Main can be run and tested before B is written; the stub is replaced by the real module later

A stub stands in for a module that is not written yet, so the modules above it can be tested now

> **Worked example.** After the program passed its in-house tests it was given to a group of users to try before release. Name this type of testing, and state what happens next.

Beta testing – real users in their own environment, reporting faults the developers did not find. The faults are corrected, then the customer carries out **acceptance**

testing against the requirements and the program is released; faults found in live use are then handled by corrective maintenance.

> **Worked example.** Give three benefits of testing a program by walkthrough.

Errors are found by people who did not write the code and so read it without assumptions; the logic is checked against the design and specification, not only against test data; several people learn how the code works, which helps later maintenance; and no test data or working computer is needed, so it can be done early.

Test strategy and test plan

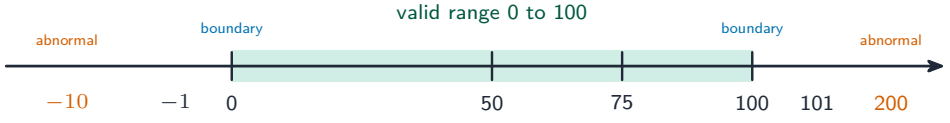
A **test strategy** is the **high-level approach** – which kinds of testing, who does them, when, and the criteria to move on. A **test plan** is the **detailed list of tests** – each with input data, expected output, and a column for the actual output.

What each contains. A *test strategy* states which testing methods will be used at which stage (module testing by the programmer, then integration, alpha, beta, acceptance), who is responsible for each, what test data is required, and the criteria for passing to the next stage. A *test plan* lists the individual tests: for each, the module or feature under test, the input data, the **reason** the data was chosen (normal, abnormal, extreme, boundary), the **expected** result, a space for the **actual** result, and what to do if they differ. The plan is written at the **design** stage, from the specification, so that it tests what the program *should* do rather than what it happens to do.

Choosing test data

For each field or condition, include three kinds:

- **normal data** – typical values inside the valid range (for marks 0–100: `50`, `75`).
- **abnormal data** – values that should be rejected (`-10`, `200`, `"abc"`).
- **extreme data** – the largest and smallest values still **accepted** (`0` and `100`).
- **boundary data** – values at the edges, where off-by-one errors hide (each accepted extreme and the rejected value just outside it: `0`/`-1`, `100`/`101`).



normal (50, 75) inside; extremes (0, 100) the accepted edges; abnormal (−1, 101, −10, 200) rejected

Test data for a 0–100 field: normal inside, extremes at the boundaries, abnormal outside

Worked example. A field accepts an exam mark from 0 to 100. Give test data of each kind with its expected result. **Normal**: `50` - accepted, a typical value inside the range. **Abnormal**: `-10`, `200`, `"abc"` - all rejected, being out of range or the wrong data type. **Extreme**: `0` and `100` - the largest and smallest values that are still **accepted**. **Boundary**: the pairs straddling each edge - `-1` rejected alongside `0` accepted, and `100` accepted alongside `101` rejected. Every value must carry its **expected result**, or the test plan proves nothing. Extreme and boundary are the pair most often confused: an extreme value sits **inside** and is accepted, while a boundary test is always a **pair** either side of the edge - which is exactly where off-by-one errors hide.

Worked example. A component passes if its weight, measured to the nearest gram, is within 3 g of the target of 50 g, i.e. from 47 g to 53 g inclusive. Draw up the test-plan rows for the check.

Test data	Type	Reason	Expected result
50	normal	a typical value well inside the range	accepted
47, 53	extreme (boundary)	the smallest and largest values that must still be accepted	accepted
46, 54	boundary	the values just outside the range, where an off-by-one error would accept them	rejected
20, 90	abnormal	values far outside the range	rejected
“abc”, −5	abnormal	the wrong type, a negative weight	rejected

Each row must say **why** the value was chosen and **what should happen**; a bare list of numbers earns nothing.

Maintenance

Most of a program’s lifetime cost is in maintenance. Three kinds:



Three kinds of maintenance: perfective, adaptive and corrective

- **perfective maintenance** – improving performance or features even though it works (a faster query, a new option).

- **adaptive maintenance** – keeping it working in a changing environment (a new OS, a new API, a legal change).
- **corrective maintenance** – fixing bugs found in use.

A program may need all three throughout its life.

Why each is needed – the reasons the mark scheme lists. *Corrective*: a fault is reported by a user after release, or an incorrect output is noticed in particular circumstances that testing did not cover. *Adaptive*: the operating system, hardware or browser is upgraded; a law or company rule changes (tax rates, data-protection requirements); the program must work with a new external system or file format. *Perfective*: users ask for extra features or a better interface; the program is made faster or made to use less memory; the code is tidied to make future changes easier.

> **Worked example.** (a) A released program outputs a wrong value under certain circumstances. (b) The hardware that runs a program is replaced. (c) Customers ask for the coffee-shop loyalty program to send a message on a customer's birthday. Name the maintenance type in each case.

(a) **Corrective** – a fault in the delivered program is being fixed. (b) **Adaptive** – the program is changed to run in its new environment. (c) **Perfective** – a feature is added to a program that already works.

Amending an existing program

When asked to add a feature or fix a bug:

1. **read the existing code** until you understand the algorithm and data flow.
2. **find where the change goes** – which subroutine, which lines.
3. **make the change as small as possible** – don't rewrite working code.
4. **update related parts** – every caller of a changed parameter list, every routine using a changed data structure.
5. **test** the new behaviour **and** the old (**regression testing** – check you broke nothing).
6. **document** the change.

Clear comments, meaningful names, decomposed subroutines and a structure chart make a program much easier to amend – which is why the design tools matter even after the first release.

Analysing a program you did not write. Start from the identifier table and the module headers: they tell you what each module receives and returns before you read a line of its body. Then trace the algorithm with a trace table for one small input, noting where each output value comes from. Only then decide where the enhancement goes – usually a new module called from the existing one, so the working code is disturbed as little as possible – and write the pseudocode for the change and the test data that proves it.

Definitions the examiner accepts

A definition question is marked against fixed wording. Learn these exactly, and give one answer only.

Term	Definition
development life cycle	the sequence of stages, from analysis to maintenance, followed to produce and support a program
waterfall model	a life cycle in which the stages are carried out in a fixed order, each completed before the next begins
iterative model	a life cycle in which a working version is produced and then repeatedly refined until it is complete
rapid application development	a life cycle that builds prototypes quickly, refining them with user feedback until they are accepted
structure chart	a diagram that shows how a program is decomposed into modules, the order in which they are called and the parameters passed between them
state-transition diagram	a diagram that shows the states a system can be in and the inputs that cause it to move between them
syntax error	an error in the way a statement is written, so it breaks the rules of the language and cannot be translated
logic error	an error in the algorithm, so the program runs but produces the wrong result
run-time error	an error that occurs while the program is running, such as division by zero, and stops it
dry run	working through the algorithm by hand, recording the values of the variables in a trace table
walkthrough	a review in which the author steps through the code with colleagues who look for errors
stub	a placeholder module with the correct header that returns a fixed value, used so the modules that call it can be tested
test plan	a list of the tests to be carried out, each with its test data, the reason for the data and the expected result
boundary data	values at each edge of the valid range, both the last value accepted and the first value rejected

Term	Definition
corrective / adaptive / perfective maintenance	fixing faults found in use / changing the program to suit a changed environment / improving a program that already works

Exam tips

- Compare development models (waterfall, iterative, RAD) by **principle, benefit, drawback**, and know the five stages of the **program development life cycle** and what each produces.
- Distinguish **syntax, logic and run-time** errors by *when* each shows itself: at translation, in the output, during the run.
- Choose **test data** of every kind – normal, abnormal, extreme and boundary – and give each value with its reason and expected result.
- Distinguish the types of **maintenance** (corrective, adaptive, perfective) by *why* the change is being made.
- On a structure chart, name every symbol: box, calling line, data couple, control couple, selection diamond, iteration arrow. Reading module headers off a chart, remember a function has `RETURNS`.

Common mistakes

- Describing a life cycle stage by its name only ("in the design stage the program is designed"). Say what is produced: structure chart, pseudocode, test plan.
- Calling a wrong output a "run-time error". If the program runs to the end, it is a logic error.
- Giving boundary data as just the extremes. The mark needs the values on **both** sides of the edge.
- Treating alpha and beta testing as the same. Alpha is in-house by the developers; beta is by real users outside.
- Confusing adaptive and perfective maintenance. Adaptive responds to a change *outside* the program; perfective improves a program nobody had to change.
- Drawing a structure chart with the modules in any order. They read left to right in the order they are called, and each parameter needs its arrow.

Basic economic ideas and resource allocation

A-Level Economics

The fundamental economic problem

Economics starts with one big idea: **scarcity** . We never have enough to give everyone everything they want.

- **resources** are the things we use to make goods and services. They are **finite** – there is only a limited amount, so they are **scarce** .
- **goods** (things you can touch, like food) and **services** (work done for you, like a haircut) are what we make from resources.
- human **wants** are unlimited. People always want more.

Because wants are bigger than resources, this is called the **economic problem** . Every society must answer three questions:

1. **What** to produce?
2. **How** to produce it?
3. **For whom** to produce it?

Choice and opportunity cost

Because resources are scarce, you must choose. Every choice means you give up something else.

opportunity cost – the next best **alternative** you give up when you make a decision.

Opportunity cost is faced by everyone:

- **consumers** (people who buy and use goods): buy a phone, and you give up the holiday that money could have paid for.
- **producers** and **firms** (the businesses that make goods): use a factory to make cars, and you give up the trucks it could have made.
- **governments** : spend tax money on hospitals, and you give up the schools that money could have built.

Thinking like an economist

Economics as a social science

Economics is a **social science** . It studies how people behave when they make choices. Like other sciences, it builds **models** (simple pictures of the real world) and tests them with data.

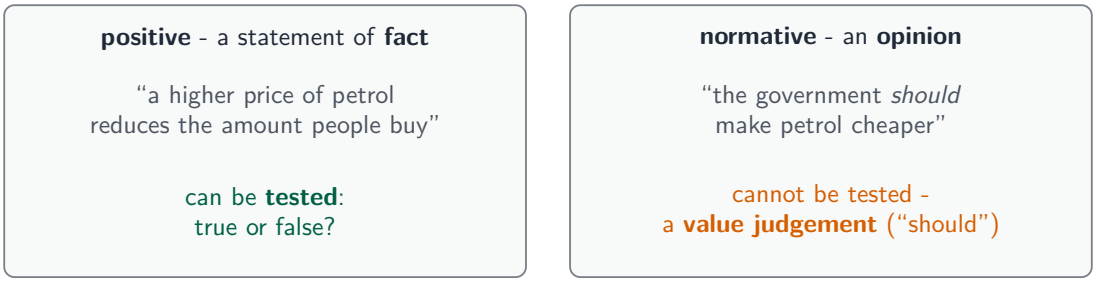
The real world is complex, so economists change one thing at a time and keep every-

thing else fixed. This is the idea of **ceteris paribus** – a Latin phrase meaning "all other things stay the same". For example: "if the price of coffee rises, people buy less coffee, *ceteris paribus*".

Positive and normative statements

- a **positive statement** is about fact. It can be tested as true or false. Example: "a higher price of petrol reduces the amount people buy."
- a **normative statement** is about opinion. It says what *should* happen, and it cannot be tested as true or false. Example: "the government should make petrol cheaper."

Normative statements rest on a **value judgement** – a personal view about what is good or fair. Two people can agree on the facts and still disagree about what *should* be done.



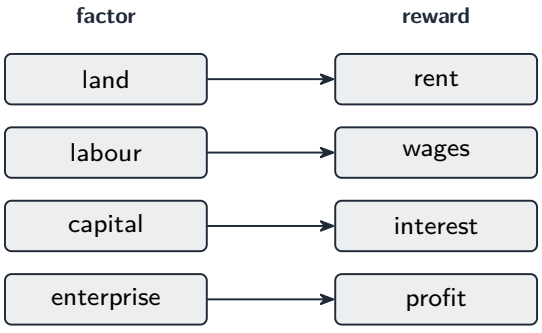
A positive statement can be tested; a normative statement is a value judgement

Microeconomics and macroeconomics

- **microeconomics** studies single parts of the economy: one market, one firm, one consumer.
- **macroeconomics** studies the whole economy together: total **output**, jobs, and rising prices.

Factors of production

The **factors of production** are the four kinds of resource used to make goods and services. Each one earns a **reward**.



The four factors of production and the reward each earns

Factor	What it is	Its reward
land	natural resources (fields, water, oil, minerals)	**rent**
labour	human work, both physical and mental	**wages**
capital	man-made things used to produce, like machines, tools and factories	**interest**
enterprise	the skill of bringing the other three together and taking the risk	**profit**

Land includes some **non-renewable resources** (like oil and coal) that can run out. Enterprise is supplied by the **entrepreneur**, the person who has the business idea and risks their own money.



An open-pit copper mine: land includes the natural resources (minerals, oil, water) that firms extract.

Specialisation and division of labour

specialisation is when a worker, firm or country makes only the things it is best at, then trades for the rest.

Inside a workplace, specialisation leads to the **division of labour**: a big job is split into small tasks, and each **worker** does one task again and again.

Advantages	Disadvantages
workers get faster and better at one task	doing one task is boring, so workers may quit
less time wasted moving between jobs	if one worker is absent, the whole line can stop
output per worker rises, so costs fall	workers learn only one narrow skill



A car assembly line: capital (machines and robots) plus the division of labour, each worker repeating one task.

Mobility of factors

mobility means how easily a factor can move to a new use.

- **occupational mobility** – how easily a factor moves from one *job or use* to another. A teacher who retrains as a nurse is occupationally mobile.
- **geographical mobility** – how easily a factor moves from one *place* to another. Land cannot move at all, so it has no geographical mobility.

Labour is often not very mobile. Family ties, the cost of housing, and missing skills all make it hard for workers to move.

Economic systems

An economic system is the way a country decides how to **allocate** (share out) its scarce resources. There are three types.

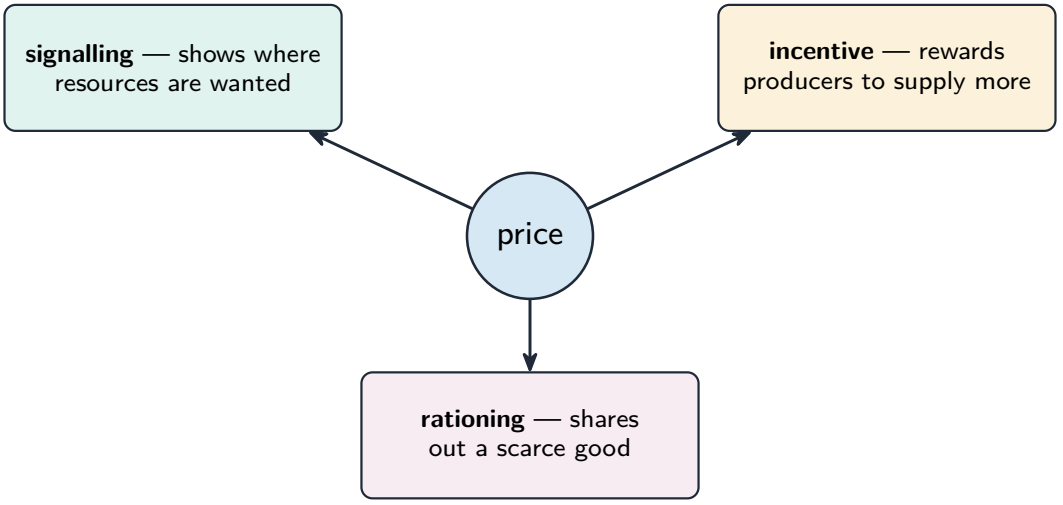
- a **market economy** : decentralised buyers and sellers influence production and allocation through prices and purchasing decisions.
- a **planned economy** (also called a **command economy**): central decisions determine resource allocation and production, often using quantities and eligibility rules.
- a **mixed economy** – a blend of the two. The **private sector** (firms and households) and the **public sector** (the government) both make decisions. Almost every real country is mixed.

Worked allocation comparison. Three illustrative clinic arrangements provide the same kind of service. Private clinics responding to paying patients illustrate market allocation: prices, demand and expected costs influence what and how much to supply, and ability to pay affects access. A central authority specifying quantities, staffing and eligibility illustrates planning. Public tax-funded appointments alongside privately sold appointments illustrate a mixed arrangement. Public provision of one service does not make the whole economy planned. Planning and tax funding do not remove scarce staff time or the need to choose who receives appointments. Ownership alone does not prove efficiency or fairness.

■ The price mechanism

In a market economy, resources are guided by the **price mechanism** . Prices do three jobs:

- **signalling** – a price tells buyers and sellers where resources are wanted. A rising price says "this is wanted – make more".
- **incentive** – a higher price rewards producers, so they want to supply more of that good.
- **rationing** – when a good is scarce, a higher price shares it out to those who are willing and able to pay.

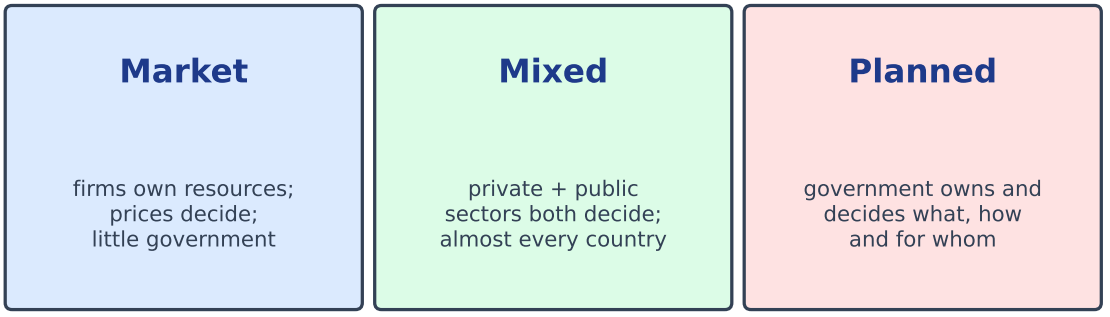


In a market the price does three jobs at once – it signals, it gives an incentive, and it rations the scarce good

Comparing the systems

System	Strengths	Weaknesses
market	choice, low prices, strong incentive to work and invent	large gap between rich and poor; some goods not supplied
planned	smaller gap between rich and poor; key goods provided for all	weak incentive; shortages and surpluses; slow to change
mixed	tries to keep the good points of both	the right balance is hard to find

In practice no country sits at a pure extreme. China's economy, for instance, is officially a *socialist market economy*, where markets set most prices while the government guides key sectors and owns major enterprises. Every real system mixes the two and sits somewhere on the spectrum below.



Exam: real economies are mixed — the question is how far market vs government decides

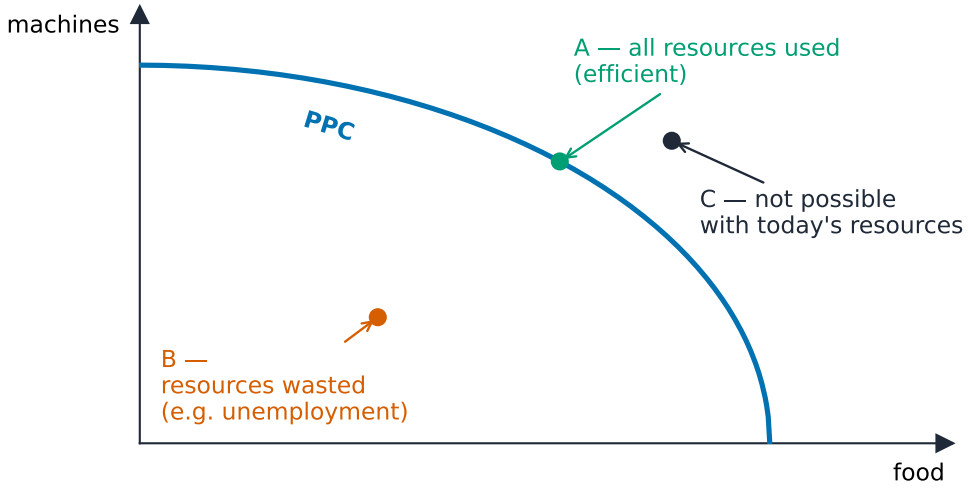
Economic systems form a spectrum: a pure market at one end, full government planning at the other, mixed economies in between.

Production possibility curves

A **production possibility curve** (PPC) is a line showing the largest combinations of two goods an economy can make when all its resources are fully and well used.

Imagine an economy that makes only food and machines. The PPC is drawn with food on one axis and machines on the other.

- a point **on** the curve: production is **efficient** for the given resources and technology. This alone does not establish the mix society prefers.
- a point **inside** the curve: resources are wasted (for example, **unemployment** of workers). The economy could do better.
- a point **outside** the curve: impossible today, because there are not enough resources. This shows scarcity.



on = efficient · inside = inefficient · outside = unattainable

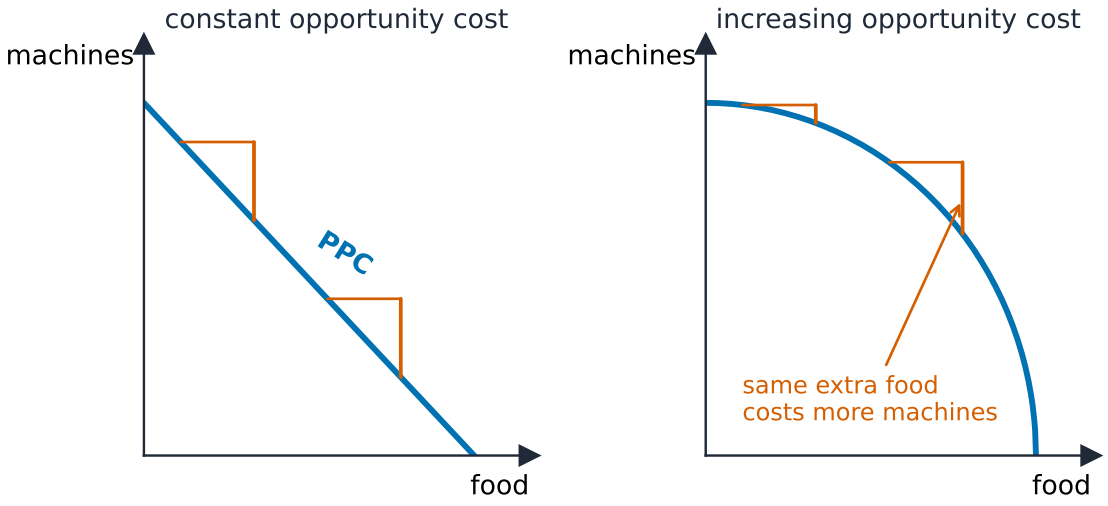
A point on the curve uses all resources (efficient); inside the curve means waste, e.g. unemployment; outside is not possible yet.

Worked frontier classification. An illustrative frontier joins food/machine combinations A=(20,0), B=(16,4), C=(10,8), D=(0,12), with food in tonnes/day and machines in units/day. E=(8,4) is inside: B permits more food at the same machine output. F=(15,8) is outside: at eight machines the maximum food is ten tonnes. C is efficient in production, but the frontier alone cannot show whether society prefers C to B. Moving from B to C gives up six tonnes/day for four extra machines/day, an interval average of 1.5 tonnes per machine. Unused capacity can permit more output without sacrificing the other plotted good; it does not make every use of scarce time, money or staff costless. Opportunity cost is not confined to points on a PPC.

Moving along the curve shows opportunity cost: to make more food, you must move resources away from machines, so you make fewer machines. This is the **trade-off** between the two goods.

The *shape* of the curve shows how opportunity cost behaves:

- a straight-line PPC means **constant** opportunity cost – each extra unit of food costs the same number of machines.
- a curve that bows outwards means **increasing** opportunity cost – each extra unit of food costs more and more machines, because resources are not equally good at making both goods.

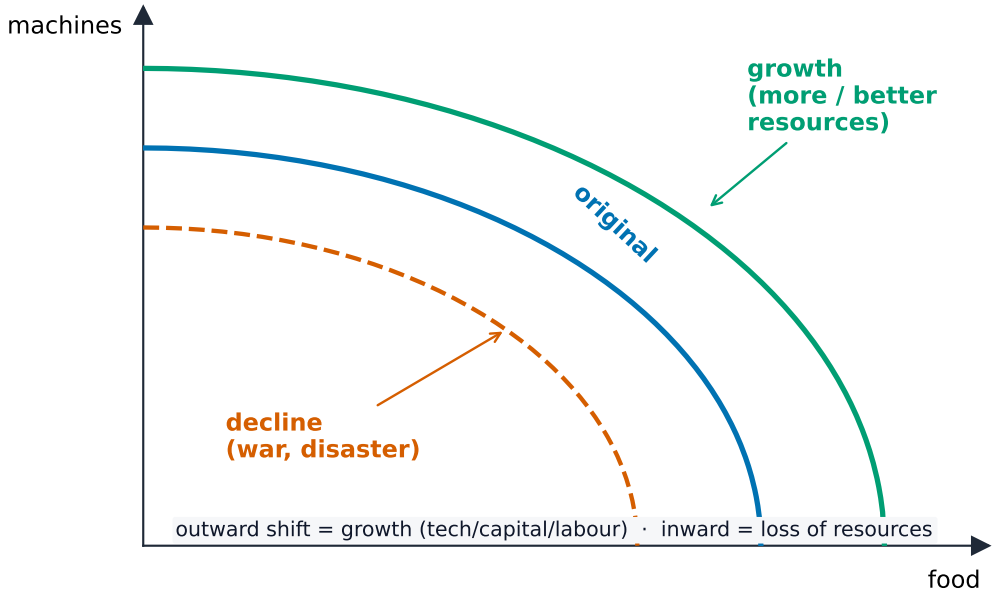


opportunity cost = goods forgone · bowed-out PPC ⇒ increasing opportunity cost

A straight PPC means each extra unit of food costs the same; a bowed PPC means each extra unit costs more and more machines.

Shifts and growth

When the PPC moves outward, productive capacity increases: this is potential **economic growth** . Actual output need not immediately rise to the new frontier. A sector-specific improvement can rotate one part of the frontier rather than shifting both intercepts. It happens when there are more resources, or better ones (new **technology** , a bigger or more skilled workforce). War or disaster can move the PPC inward – this is **economic decline** .

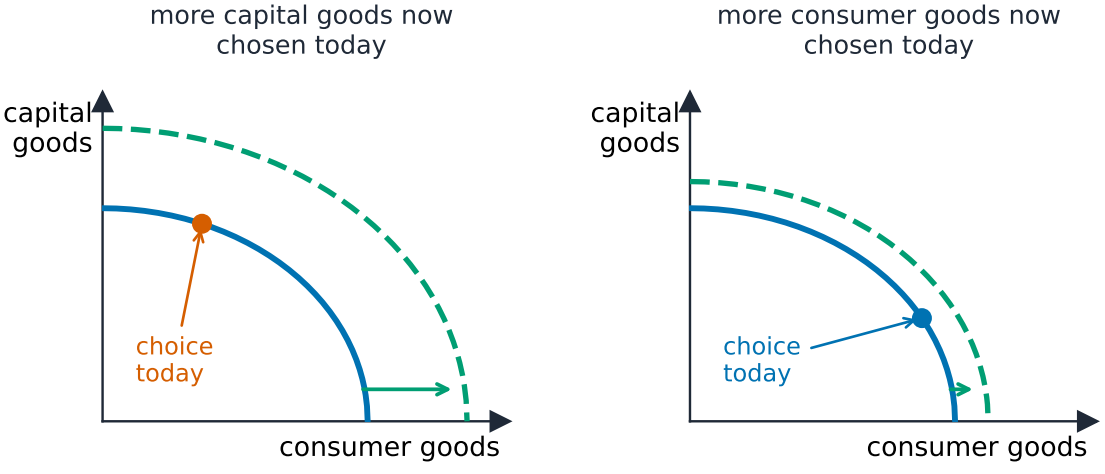


More or better resources shift the whole PPC outward (growth); war or disaster shifts it inward (decline).

A country must also choose between two kinds of good now:

- **capital goods** – goods used to make other goods (machines, factories).
- **consumer goods** – goods people use directly (food, clothes).

Making more capital goods today means fewer consumer goods today. Productive investment can increase future capacity, provided it adds effective resources after wear and replacement. More capital spending does not guarantee a larger shift: quality, use, skills and other constraints matter.



Blue is the current PPC; dashed green is an illustrative future PPC. The larger future shift assumes the extra investment adds productive capacity.

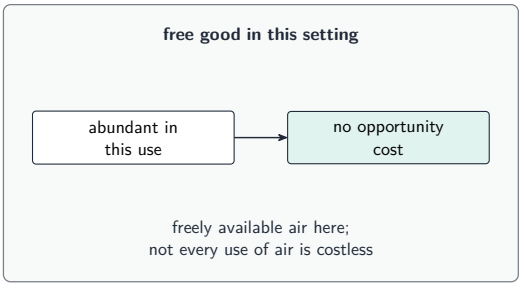
Worked example. Why is a production possibility curve usually drawn **concave** (bowed outwards) rather than as a straight line? A straight line would mean the opportunity cost of each extra unit stays **constant**. The curve bows out because factors of production are **not perfectly substitutable** - they are specialised. Starting from all-food production, the first factors moved into making cars are the ones **least suited** to farming, so very little food is given up for a lot of cars. Keep going and you must move factors that are **increasingly good at farming and increasingly bad at making cars**, so each extra car costs **more and more** food. That rising cost is **increasing opportunity cost**, and the bowed shape is simply its picture. A straight-line PPC is not wrong - it just assumes factors are perfectly substitutable, and naming that assumption is what earns the mark.

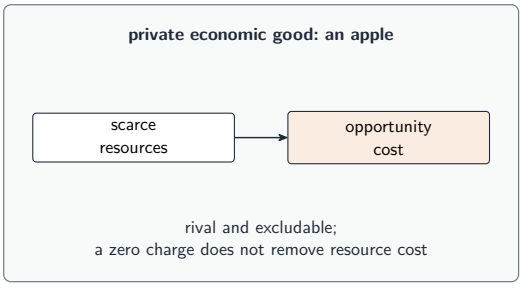
Types of goods and services

Free goods and economic goods

- a **free good** has no opportunity cost in the relevant use: available supply is sufficient without sacrificing another valued use. Freely available air may qualify in one setting; supplying clean air or oxygen in a hospital can require scarce resources. A zero charge alone does not prove a good is free in this economic sense.
- an **economic good** uses scarce resources and has opportunity cost. A **private good** has rivalry and excludability. An apple is an example; economic resources can also be used to provide public services. A good can have opportunity cost even if its user is not directly charged.

A private good has two features. It has **rivalry** (if I eat the apple, you cannot) and **excludability** (the seller can stop you using it unless you pay).





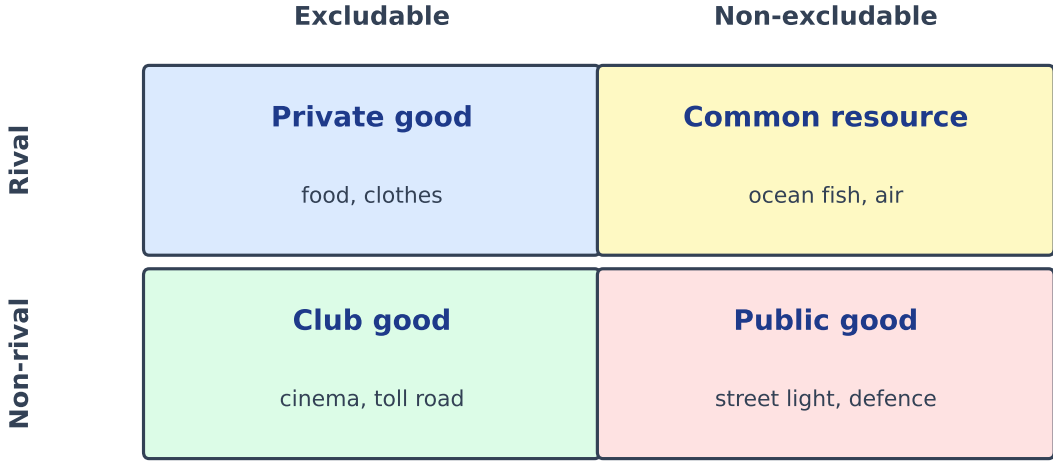
Opportunity cost depends on scarcity in the relevant use. A private economic good also has rivalry and excludability; a zero charge does not establish zero opportunity cost.

Public goods

A **public good** is the opposite of a private good. It has two features:

- **non-rivalry** – one person using it does not reduce the amount left for others.
- **non-excludability** – you cannot stop people who do not pay from using it.

Because of non-excludability, public goods suffer from the **free-rider problem**: people enjoy the good without paying for it. This can cause private under-provision because voluntary payments may not cover cost. It does not prove nobody contributes: donations, sponsorship or other funding can support some provision. Government financing is one response; the financing body and the actual provider need not be the same. Street lighting and **national defence** are standard examples under the non-rival, non-excludable assumptions.



Exam: classify by rivalry × excludability — free-rider problem only for non-excludable goods

Rivalry and excludability give four types of good: private, club, common resource, and public.

Worked classification. An uncrowded local street light benefits nearby users and cannot readily exclude non-payers, fitting the public-good conditions in that setting. A publicly funded hospital appointment uses a scarce doctor's time even when the patient pays nothing. It can be reserved for a patient and cannot simultaneously serve everyone, so public ownership does not make it a public good. A paid lesson can be private and merit at once if the learner underestimates its benefit; the information test differs from rivalry and exclusion.

Merit and demerit goods

- a **merit good** is better for you (and for society) than you realise, so people buy too little of it. Examples: education and health care.
- a **demerit good** is worse for you than you realise, so people buy too much of it. Examples: cigarettes and alcohol.



Education is a classic merit good. Its full value – better jobs, and a more skilled society – is easy to underestimate, so left to themselves people buy less of it than is good for them or the country. That is why governments provide and subsidise it

People misjudge these goods because of **information failure** – they do not have, or do not use, the full facts about the benefit or harm.

Deciding what counts as a merit or demerit good is partly a value judgement, so people may disagree about which goods belong on each list.

Exam tips

- Frame answers around **scarcity and choice**: every choice has an **opportunity cost** – the next best alternative given up.
- Read a **PPC**: a point inside means unused resources, on the curve is efficient, outside is unattainable; the slope shows opportunity cost.
- Distinguish **positive** (a testable fact) from **normative** ("should", a value judgement) statements.
- Classify goods: free versus economic, and **public / merit / demerit** goods (non-excludable, non-rival, and so on).

Vocabulary card meanings

goods: Physical items that people can buy and sell. alternative: Any other option a person could have chosen. social science: A field that studies how people behave

in groups and markets. ceteris paribus: Holding every other factor unchanged while one factor is studied. positive statement: A claim that can be checked against facts. normative statement: A claim about what ought to happen, resting on a value judgement. microeconomics: The study of single markets, firms, and households. reward: The payment a factor of production receives for its use. rent: The payment made for the use of land. non-renewable resources: Inputs that nature does not restore within a human lifetime after they are used. division of labour: Splitting production so that each person does one task. worker: A person who supplies labour. mobility: How easily a factor can move to another job or place. geographical mobility: How easily a person can move to work in another place. planned economy: An economy in which the state decides what is produced. command economy: An economy run by government orders rather than by prices. economic decline: A fall in production and incomes across the whole economy. national defence: Protection of a country by its armed forces.

The price system and the microeconomy

A-Level Economics

Demand

demand is the quantity of a good that buyers are willing and able to buy at each **price** in a period of time. It is not just a wish – buyers must also be able to pay.

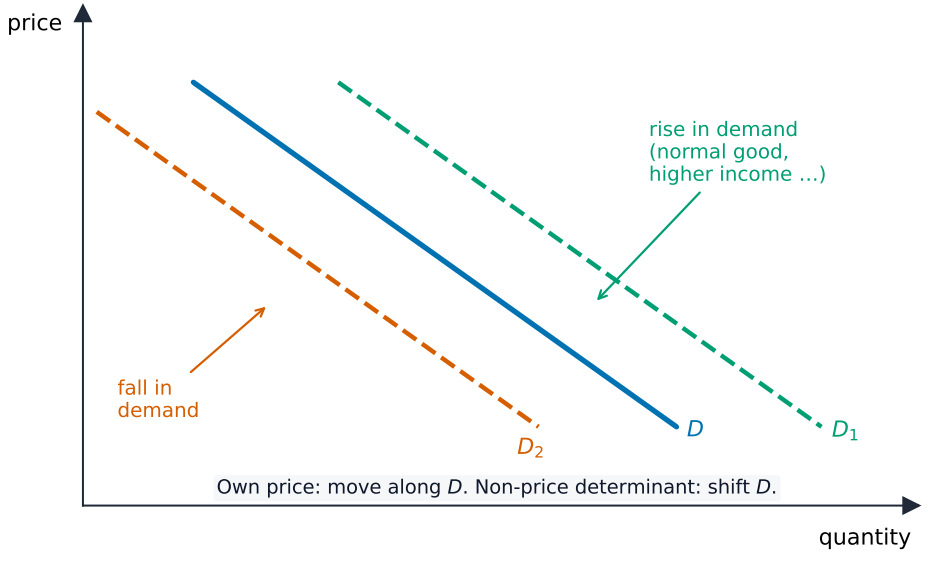
The **law of demand**, holding other determinants constant, says: when the price rises, the **quantity demanded** falls; when the price falls, the quantity demanded rises. Price and quantity move in opposite directions.

So the **demand curve** slopes downward, from top-left to bottom-right.

What shifts the demand curve

A change in the good's **own price** is shown as a *movement along* the demand curve. A change in a non-price determinant shifts the whole curve to a new position – this is a **shift** of demand. The main causes (the **determinants**) are:

- **income** – for most goods, higher income raises demand.
- the price of a **substitute** – a good you can use instead (tea for coffee). If the substitute's price rises, demand for this good rises.
- the price of a **complement** – a good used together with this one (cars and petrol). If the complement's price rises, demand for this good falls.
- tastes and fashion, the size of the population, and advertising.



A change in income, tastes or related prices shifts the whole curve (D_1 right, D_2 left); a change in the good's own price is a movement along it.

Apply movement versus shift. Consider three separate coffee-market events. (1) Coffee price falls from \$7 to \$6, all other conditions fixed: quantity demanded rises along the original curve. (2) Tea becomes dearer: coffee demand shifts right **if** tea is a substitute. (3) Income rises, but no income relationship is given: the direction needs a normal/inferior-good assumption. Draw the relevant change before reading each explanation.

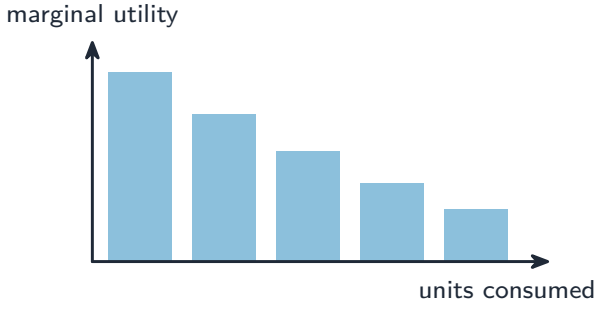
■ Individual and market demand

One **consumer** has an individual demand. The **market demand** is found by adding up the quantity that all consumers want at each price (adding the curves sideways).

■ Marginal utility and demand

utility is the satisfaction you get from consuming a good. **marginal utility** is the extra utility from one more unit.

The law of **diminishing marginal utility** says each extra unit gives less extra satisfaction than the one before. This can help explain lower willingness to pay in a simple model, but it is not by itself a general proof of downward-sloping demand. Full utility analysis is an A-Level Topic 7 preview here. For marginal utilities $12, 9, 6, 3$, total utility is $12, 21, 27, 30$: total utility rises while marginal utility stays positive.



Each extra unit adds less satisfaction in this illustrative diminishing-marginal-utility model

Supply

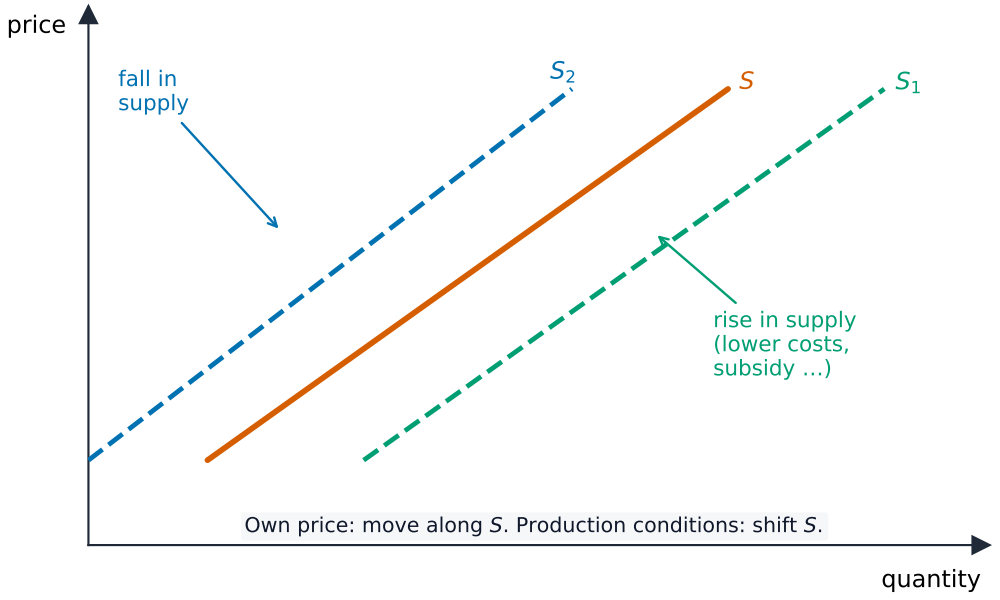
supply is the quantity of a good that **producers** are willing and able to sell at each price in a period of time.

The **law of supply**, holding other determinants constant, says: when the price rises, the **quantity supplied** rises. Higher prices can make extra output worthwhile. Total **profit** also depends on costs; price alone does not establish it. So the **supply curve** slopes upward.

What shifts the supply curve

A change in the good's own price is a movement along the supply curve. Other causes shift the whole curve:

- **costs of production** – if wages, raw materials or rent get dearer, supply falls.
- technology – better machines raise supply.
- a per-unit producer **subsidy** raises supply; a per-unit tax on sellers lowers it.
- the number of **firms** in the market, and weather (for farm goods).



Lower costs, better technology or a subsidy shift supply right (S_1); higher costs or a tax shift it left (S_2).

Bakery application. A higher bread price, other conditions fixed, raises quantity supplied along the original curve. Cheaper flour or a more efficient oven shifts supply right: more bread can be offered at the **same** price. A change unrelated to willingness or ability to produce does not automatically shift supply.

Individual and market supply

The **market supply** is found by adding up the quantity all firms will sell at each price.

Price elasticity of demand

price elasticity of demand (PED) measures how much the quantity demanded responds when the price changes.

$$\text{PED} = \frac{\%\text{ change in quantity demanded}}{\%\text{ change in price}}$$

PED is normally negative (because of the law of demand), but we usually look at the size and drop the minus sign.

> **Worked example.** The price of a good rises by 10% and the quantity demanded falls by 25%. Find the PED and classify the demand.

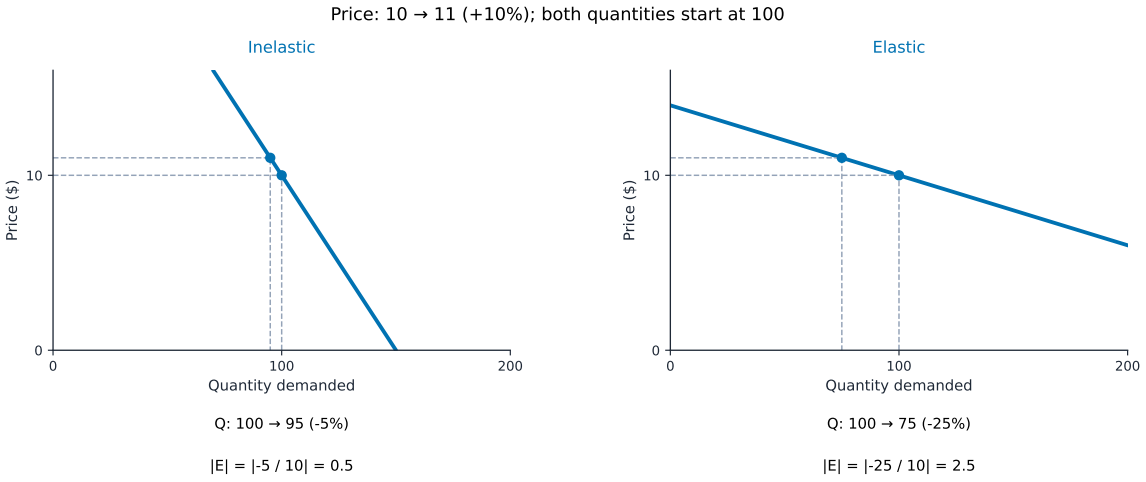
$$\text{PED} = \frac{-25}{10} = -2.5$$

Ignoring the minus sign, $2.5 > 1$, so demand is elastic – quantity responded much more than the price did.



A petrol station: demand responsiveness depends on available alternatives and adjustment time; the photo alone cannot establish PED.

PED magnitude	Name	Meaning
more than 1	**elastic**	quantity responds a lot
less than 1	**inelastic**	quantity responds a little
equal to 1	**unit elastic**	quantity changes by the same % as price
0	**perfectly inelastic**	quantity does not change at all
infinity	**perfectly elastic**	horizontal demand at a given price in the idealized model



The same 10% price rise gives quantity falls of 5% and 25%: PED magnitudes 0.5 and 2.5. Compare percentages, not slope alone.

What makes demand elastic or inelastic

- the number and closeness of substitutes (more substitutes → more elastic).
- whether the good is a **necessity** or a **luxury**; these influence responsiveness but do not guarantee a particular own-price coefficient.
- the share of income spent on the good (a large share → more elastic).
- time – more time usually lets buyers find alternatives, making demand more responsive; it does not guarantee elasticity above 1.
- habit or addiction can reduce responsiveness, alongside other determinants.

One straight-line curve. For $Q_d = 10 - P$, the gradient $\Delta Q_d/\Delta P = -1$ is constant. **Point elasticity** uses that gradient with the current price and quantity:

$$PED = \frac{\Delta Q_d}{\Delta P}\frac{P}{Q_d}.$$

At $(P, Q) = (8, 2)$, $PED = (-1)(8/2) = -4$; at $(5, 5)$, it is -1; at $(2, 8)$, it is -0.25. The curve has the same slope but different percentage responsiveness. These are point values, not a claim that a large finite change has the same elasticity. Revenue $TR = PQ$ is $16, 25, 16$ respectively: the midpoint maximizes revenue on this straight-line curve with positive intercepts. A constant unit-elastic demand curve instead keeps revenue constant throughout its unit-elastic segment.

PED and total revenue

total revenue is the money a firm earns: TR = price × quantity. PED tells a firm what happens to revenue when it changes the price:

If demand is...	price rises	price falls
inelastic	total revenue rises	total revenue falls
elastic	total revenue falls	total revenue rises
unit-elastic segment	total revenue stays the same	total revenue stays the same

Worked revenue comparison. Both goods start at price \$10 and quantity 100. Price rises to \$11 (+10%, using the initial base). Good A sells 95 (−5%); good B sells 75 (−25%).

$$PED_A = \frac{-5}{10} = -0.5, \qquad PED_B = \frac{-25}{10} = -2.5.$$

$$TR_0 = P_0Q_0 = 10(100) = \$1000.$$

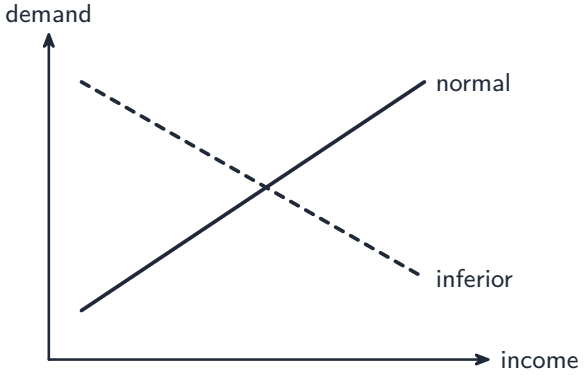
$$TR_{A,1} = P_1Q_{A,1} = 11(95) = \$1045, \qquad TR_{B,1} = 11(75) = \$825.$$

Higher revenue does not establish higher profit without cost information. **Governments** also use responsiveness when considering taxes, but tax revenue requires the tax per unit and the taxed quantity; PED alone gives neither.

Official application: 9708/21 O/N/25 Q2(a). Explain why a commuter may keep using petrol this week but change transport or car over a year. Then judge whether time is decisive: a remote commuter with no viable alternatives may still be unresponsive later. Use the formula, explain the adjustment mechanism and reach a conditional judgement; do not just list determinants.

Income elasticity of demand

income elasticity of demand (YED) measures how much demand responds when income changes.



As income rises, demand for a normal good rises but an inferior good falls

$$\text{YED} = \frac{\%\text{ change in quantity demanded}}{\%\text{ change in income}}$$

- a **normal good** has positive YED: demand rises as income rises. Necessities have YED between 0 and 1; luxuries have YED above 1.
- an **inferior good** has negative YED: demand falls as income rises (people switch to better goods). Cheap instant noodles may be inferior for some consumers over a particular income range; the label is not a judgement about quality.

Worked example. A person's income rises by 20% and their demand for restaurant meals rises by 30%. Find the YED and classify the good.

$$\text{YED} = \frac{+30}{+20} = +1.5$$

YED is positive and greater than 1, so restaurant meals are a normal good – and a luxury.

YED helps firms plan for income changes. The nature of the good, income level and whether alternatives are available can affect its value; the estimate need not remain fixed as conditions change.

Cross elasticity of demand

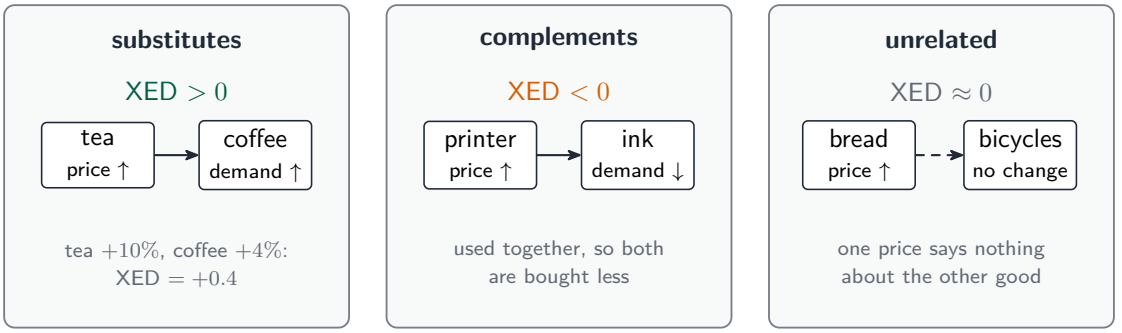
cross elasticity of demand (XED) measures how much demand for one good responds when the price of *another* good changes.

$$\text{XED} = \frac{\%\text{ change in quantity demanded of A}}{\%\text{ change in price of B}}$$

- for **substitutes**, XED is positive (the price of B rises, so people buy more A).
- for **complements**, XED is negative (the price of B rises, so people buy less A and less B).
- for goods that are not related, XED is about zero.

Worked example. When the price of tea rises by 10%, the quantity of coffee demanded rises by 4%. Find the XED and say how the two goods are related.

$$\text{XED} = \frac{+4}{+10} = +0.4$$



The sign of XED: positive for substitutes, negative for complements, about zero for unrelated goods

XED is positive, so tea and coffee are substitutes.

XED depends on the closeness of substitutes or strength of complementary use, and on the market definition. A positive value predicts that a rival price cut reduces this good's demand, other determinants fixed.

Complement practice. Printer price rises 10% and ink quantity demanded falls 6%: $XED_{ink,printer} = \frac{-6}{10} = -0.6$. The negative sign supports complementary use. It says nothing about whether ink is an inferior good; that needs an income change.

Decision practice for 9708/21 O/N/25 Q2(b). A fictional restaurant has YED = 1.5 and XED with a close rival = 0.8. In separate scenarios, income rises 2% or the rival cuts price 10%.

$$\%\Delta Q = YED(\%\Delta Y) = 1.5(2) = +3\%.$$

$$\%\Delta Q = XED(\%\Delta P_B) = 0.8(-10) = -8\%.$$

YED matters for income forecasts; XED matters for rival-price changes. Use the expected changes and business decision to judge which is more useful. These are estimates holding other determinants fixed, not guaranteed exact forecasts or an unconditional ranking of the two measures.

Price elasticity of supply

price elasticity of supply (PES) measures how much the quantity supplied responds when the price changes.

$$\text{PES} = \frac{\%\text{ change in quantity supplied}}{\%\text{ change in price}}$$

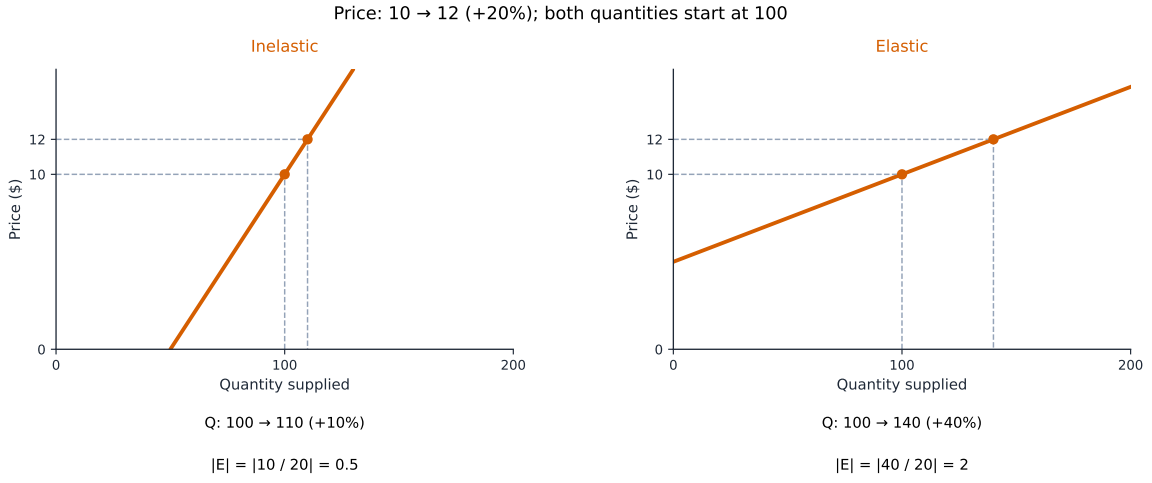
PES is nonnegative for an upward-sloping or vertical supply curve. A value above 1 is elastic; below 1 is inelastic. Compare percentage changes, not slope alone.

Compare two firms

Both start at price 10, quantity 100. Price rises to 12 (+20%). A supplies 110 (+10%); B supplies 140 (+40%). Percentages use the initial values.

$$PES_A = \frac{\%\Delta Q_s}{\%\Delta P} = \frac{10}{20} = 0.5, \qquad PES_B = \frac{40}{20} = 2.$$

Firm A is inelastic and firm B elastic. The diagram uses the same initial values and scales; visual steepness alone cannot classify arbitrary supply curves.



The same price rise: a small supply response when PES is below 1, a large one when PES is above 1

What makes supply more elastic:

- **spare capacity** – unused machines and workers, so firms can quickly make more.
- **stocks** – goods stored in a warehouse that can be sold fast.
- how easily factors of production can move into the industry.
- time. Economists split time into three periods:
  - the **momentary** period – supply is fixed; PES is zero.
  - the **short run** – at least one factor is fixed; spare capacity and stocks may still allow a substantial response.
  - the **long run** – all factors can change, usually allowing more adjustment. This does not guarantee PES above 1.

Policy application: 9708/02/SP/23 Q3(b)

Refrigeration can preserve saleable stocks; training can ease skill constraints. Neither necessarily makes **all** agricultural supply elastic: perishability, costs and biological growing time differ. Explain the constraint removed, a limitation, then a product-specific judgement. A supply shift after a subsidy is not by itself evidence that PES has increased.

The interaction of demand and supply

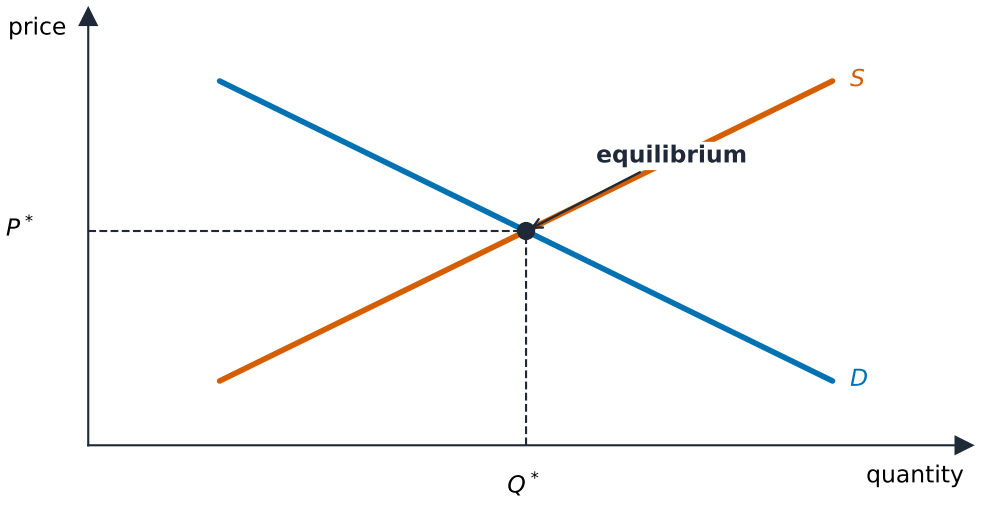
Compare the quantities demanded and supplied at the same price.



A stall offers goods at posted prices; this photograph does not establish market equilibrium

Equilibrium

The market is in **equilibrium** where the demand curve crosses the supply curve. Here quantity demanded equals quantity supplied at the same price. This gives the **equilibrium price** and the **equilibrium quantity** . There is no pressure for the price to change.



equilibrium: $Q_d = Q_s$ at (P^*, Q^*) (market clears)

At the equilibrium price P^* and quantity Q^* the curves cross, so quantity demanded equals quantity supplied.

Worked market. Let $Q_d = 12 - P$ and $Q_s = P - 2$. At prices $4, 6, 7, 8$, the corresponding demand quantities are $8, 6, 5, 4$ and supply quantities $2, 4, 5, 6$. Solve the equality:

$$Q_d = Q_s, \qquad 12 - P = P - 2, \qquad P^* = \$7, \quad Q^* = 5.$$

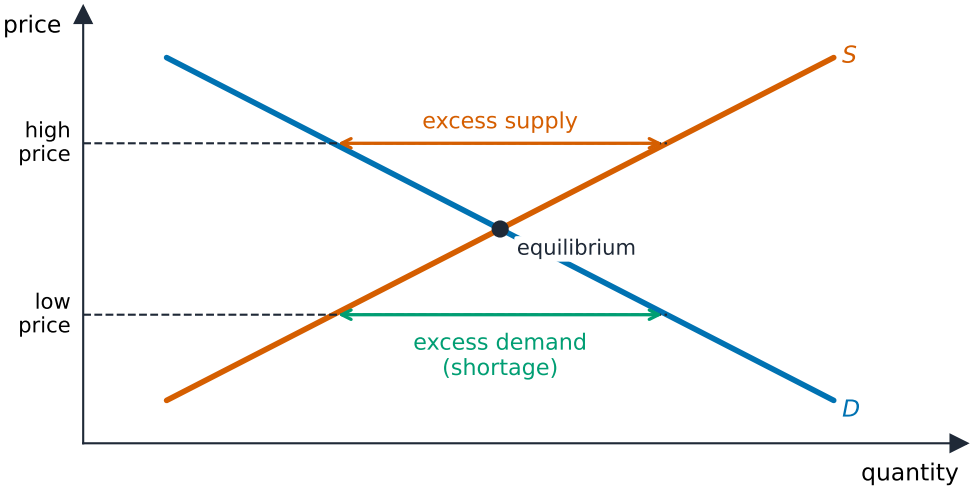
At \$6, excess demand is $6 - 4 = 2$ units; at \$8, excess supply is $6 - 4 = 2$ units. A shortage provides **signalling** of unmet demand. A higher price performs **rationing** through willingness **and ability** to pay and gives suppliers an **incentive** to offer more, other costs unchanged.

Disequilibrium

When the price is not at equilibrium, the market is in **disequilibrium** :

- if the price is **too low**, there is **excess demand** (a **shortage**). Buyers compete, so the price is pushed up.
- if the price is **too high**, there is **excess supply** . Sellers cannot sell everything, so the price is pushed down.

With other conditions unchanged and prices free to adjust, these imbalances create pressure towards equilibrium. Contracts, controls or new shocks can interrupt the path. This shows the price mechanism doing its jobs of signalling, rationing and giving an incentive.

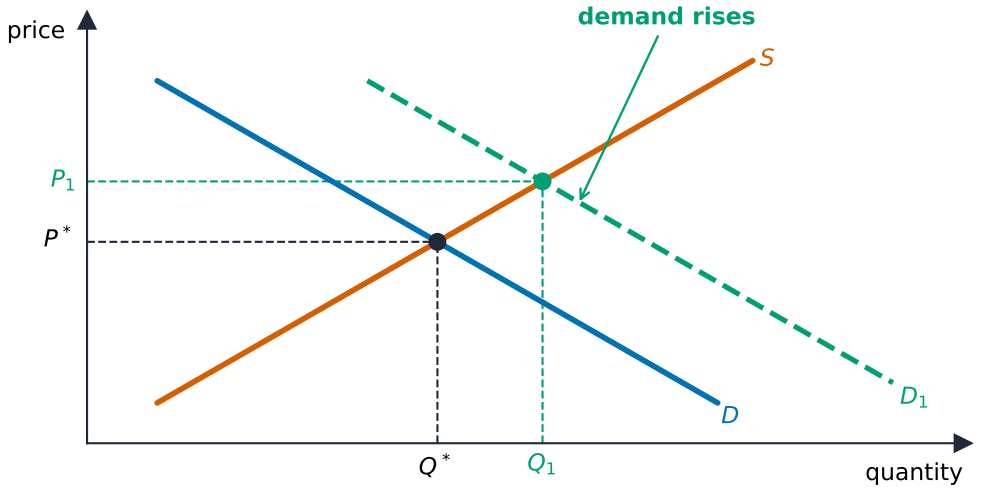


P too high ⇒ surplus · P too low ⇒ shortage · price adjusts toward P^*

Above the equilibrium price there is excess supply; below it there is excess demand (a shortage). With flexible prices and other conditions unchanged, the imbalance creates pressure towards equilibrium.

How shifts change the equilibrium

Change	Effect on price	Effect on quantity
demand rises (shifts right)	rises	rises
demand falls (shifts left)	falls	falls
supply rises (shifts right)	falls	rises
supply falls (shifts left)	rises	falls



$D \uparrow$ (S fixed) $\Rightarrow P_e \uparrow$ and $Q_e \uparrow$

With supply fixed, a rise in demand ($D \rightarrow D_1$) raises both the equilibrium price and the equilibrium quantity.

Demand and supply in different markets

The same analysis works in many markets:

- in **product markets** , demand and supply set the price of goods and services.
- in **factor markets** , they set the price of factors – for example, the demand for and supply of labour set the **wage** .
- in a floating **foreign exchange market** , the demand for and supply of a currency set the **exchange rate** (the price of one currency in terms of another).

Relationships between markets

- **Joint demand** : printers and ink are complements; more printer use can increase ink demand.
- **Alternative demand** : substitutes such as tea and coffee; dearer tea can increase coffee demand.
- **Derived demand** : bread demand creates demand for bakers' labor.
- **Joint supply** : beef production also produces hides; more beef output can increase leather supply.

Identify the market and relevant determinant before drawing its curve.

Consumer and producer surplus

consumer surplus is the gain to buyers. It is the difference between the most a consumer was willing to pay and the price actually paid. On a diagram it is the area below the demand curve and above the price line.

producer surplus is the gain to sellers. It is the difference between the price a producer receives and the lowest price the producer would have accepted. On a diagram it is the area above the supply curve and below the price line.

Worked surplus and cost change. Use the same fictional competitive market: inverse demand $P_D = 12 - Q$ and inverse supply $P_S = 2 + Q$. Equilibrium is $Q = 5$, $P = \$7$.

$$CS = \tfrac{1}{2}Q(P_{D,0} - P) = \tfrac{1}{2}(5)(12 - 7) = \$12.50.$$

$$PS = \tfrac{1}{2}Q(P - P_{S,0}) = \tfrac{1}{2}(5)(7 - 2) = \$12.50.$$

$$TS = CS + PS = \$25.$$

For 9708/22 O/N/25 Q2(a), consider the claim that producers always gain when a cost increase raises price. These are illustrative numbers, not data from that paper. Higher input costs shift supply to $P_{S,1} = 4 + Q$:

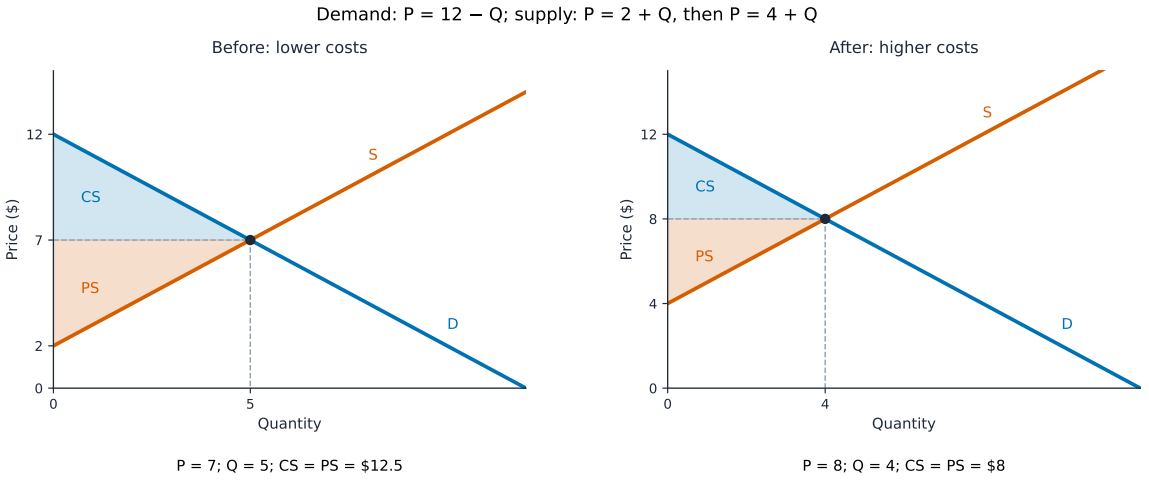
$$P_D = P_{S,1}, \qquad 12 - Q = 4 + Q, \qquad Q_1 = 4, \quad P_1 = \$8.$$

$$PS_1 = \tfrac{1}{2}Q_1(P_1 - 4) = \tfrac{1}{2}(4)(8 - 4) = \$8.$$

$$CS_1 = \tfrac{1}{2}Q_1(12 - P_1) = \tfrac{1}{2}(4)(12 - 8) = \$8.$$

$$TR_0 = P_0Q_0 = 7(5) = \$35, \qquad TR_1 = P_1Q_1 = 8(4) = \$32.$$

Price rises, but producer surplus and revenue fall in this example. The new equilibrium can still be efficient: lower total surplus reflects the higher real production costs, not automatically deadweight loss. Judge the outcome using the new cost curve, quantity and demand/supply responsiveness. Producer surplus is not automatically total profit because fixed costs also matter.



Consumer surplus is the area below demand and above the price; producer surplus is the area above supply and below the price.

A demand increase raises equilibrium price and quantity with supply fixed, but do not infer consumer surplus from price alone: both the demand curve and price have changed. A supply increase raises consumer surplus with demand fixed; producer surplus depends on the new curve and outcome. If a price is imposed away from equilibrium, specify the quantity actually traded and any rationing before measuring areas.

Consumer surplus plus producer surplus is total market **welfare** in this simple model. Competitive equilibrium maximizes it when private costs and benefits reflect social costs and benefits and there are no market failures. This is a conditional benchmark, not a claim about every real market.

Exam tips

- Quote the **elasticity formula** and interpret the value: PED magnitude inelastic (< 1), elastic (> 1); link PED to total revenue.
- Distinguish a **movement along** (a price change) from a **shift** (a non-price factor) of demand or supply.
- Use the **sign** of XED (substitutes +, complements −) and YED (normal +, inferior −).
- On a diagram, label **consumer and producer surplus** and show how a price change alters each.

Vocabulary card meanings

market demand: The total amount buyers want at each price. diminishing marginal utility: Each extra unit adds less satisfaction than the unit before it. market supply: The total amount sellers offer at each price. perfectly elastic: A situation in which price stays unchanged across the whole range of quantities buyers or sellers choose. momentary: Lasting only an instant, before anyone can adjust. equilibrium quantity:

The amount bought and sold when buyers' and sellers' plans match. product market: The place where finished goods and services are traded. point elasticity: Responsiveness of quantity, measured at one exact price. joint demand: Demand for two products that are wanted together. alternative demand: Demand for two products that can replace each other. joint supply: Two products that come from the same production process.

Government microeconomic intervention

A-Level Economics

Why markets fail

A free market often gives a good result, but not always. **market failure** is when the market does not **allocate** (share out) resources efficiently – too much or too little of a good is made. Then the government may step in.

The AS core in this topic is public goods, merit and demerit goods, price intervention, and income and wealth inequality. The externality diagrams and permit discussion below are a preview of the later A-Level microeconomics units; they do not replace the AS requirements.

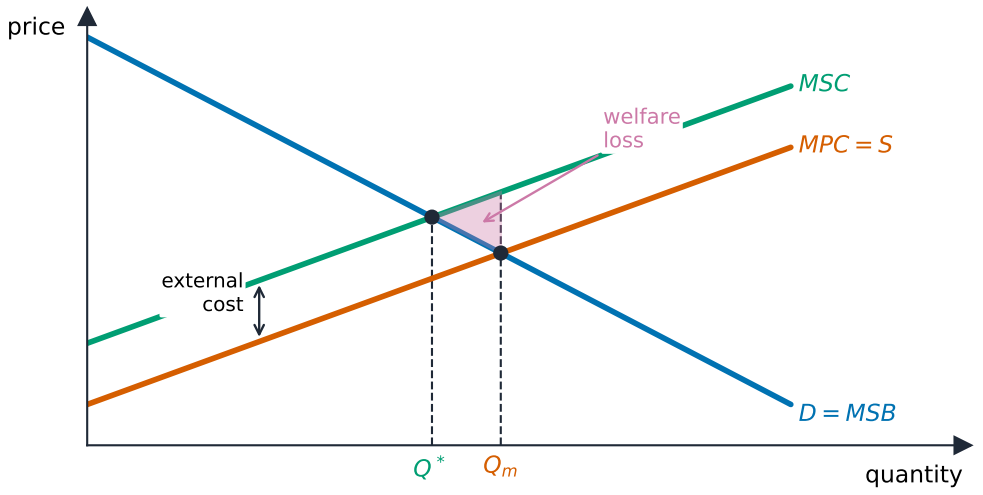


A coal power station: producing electricity can impose a negative externality such as pollution on people who are not part of the deal.

Externalities

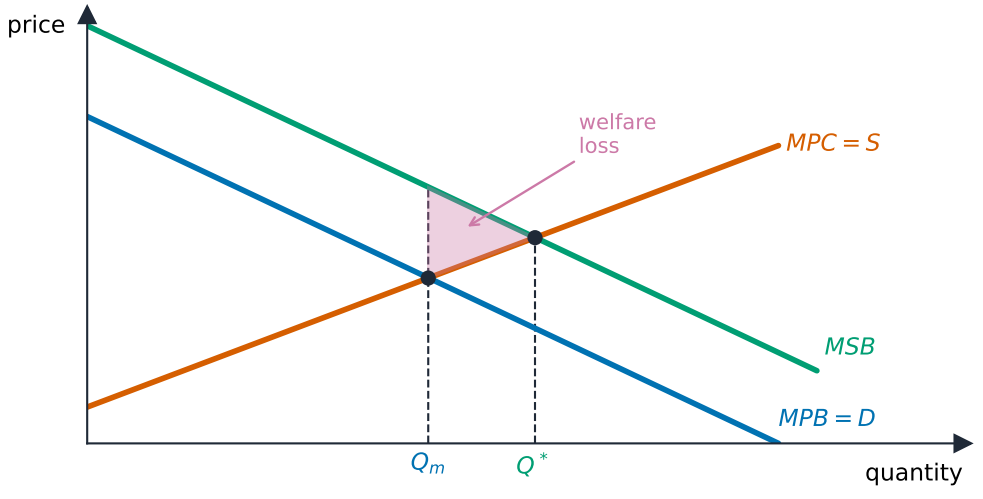
An **externality** is a cost or a benefit that falls on a **third party** – someone who is not the buyer or the seller.

- a **negative externality** is an **external cost** paid by others, like the pollution from a factory. The **social cost** (the cost to all of society) is bigger than the **private cost** (the cost to the producer). When people ignore the external cost, the market makes **too much** of the good.
- a **positive externality** is an **external benefit** enjoyed by others, like the protection that vaccination gives to everyone. Here the market makes **too little**, because buyers ignore the benefit to others.



$MSC > MPC \Rightarrow Q_m > Q^*$ · DWL from over-production

With a negative externality, social cost (MSC) is above private cost (MPC), so the market over-produces at Q_m instead of the social optimum Q^*; the shaded triangle is the welfare loss.



$MSB > MPB \Rightarrow Q_m < Q^*$
Welfare loss from under-consumption

With a positive externality, social benefit (MSB) is above private benefit (MPB), so the market under-produces at Q_m instead of the social optimum Q^*.

Public goods

A **public good** has **non-rivalry**: one more user's benefit does not reduce what existing users receive. It also has **non-excludability**: preventing non-payers from benefiting is impractical. A flood-warning broadcast to a whole valley is an example. An individual cinema seat is different: a ticket can exclude non-payers and an occupied seat is unavailable to another person. A school lesson can be a merit good without being a pure public good; places and teacher time are limited.

Worked financing problem. A warning service costs £90 for one period. Three households value it at £50, £40 and £30. Their combined benefit is £120, above its cost, but no household values it enough to pay the entire £90 alone. Once the warning is broadcast, each can benefit without paying. This **free-rider problem** can prevent an ordinary seller from recovering the cost. It is not proof that every voluntary contribution is impossible: agreement, donations or a larger organisation may fund some provision. Tax-funded provision can solve the charging problem, but government must still estimate benefits and costs and decide which service to fund.

Merit and demerit goods

A **merit good** is consumed less than is considered desirable when people underestimate or overlook its benefits. A **demerit good** is consumed more than is considered desirable when people underestimate or overlook its harm. This **information failure** concerns people's decisions about their own benefits and costs. A benefit or harm to a third party is a separate externality; one activity may involve both.

Compare two barriers. A training course charges £200. Some adults wrongly believe the qualification has no value; others understand its value but cannot pay. Information about recognised qualifications addresses the first barrier. A subsidy or free state place may address the second. If there are only 20 places, a campaign alone cannot create another 30 places. A good answer identifies the cause, explains the policy's effect on consumption, and checks resources and opportunity cost.



Cigarettes illustrate a demerit good: users may underestimate their own harm. Smoke affecting others is a separate external effect.

Monopoly power

A **monopoly** is a single seller in a defined market. **Monopoly power** is the ability to influence price rather than accept it. Restricting output can raise price and harm consumers, but costs, entry barriers and available substitutes matter. Formal efficiency and firm-behaviour analysis belongs to the later A-Level microeconomics units.

Inequality and asymmetric information

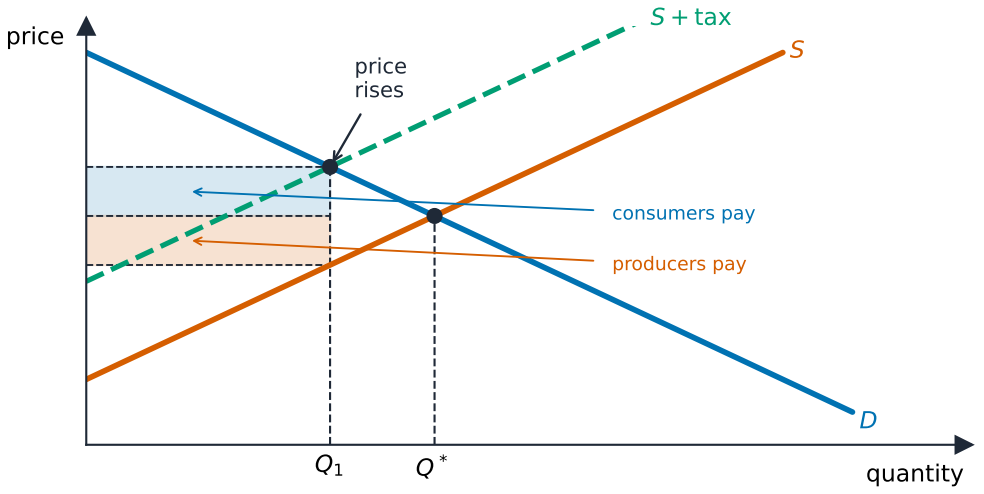
- differences in earnings and asset ownership can produce **inequality** . An unequal distribution and an inefficient allocation are different claims: inequality alone does not establish a failure of allocative efficiency.
- **asymmetric information** means one party knows relevant information the other lacks. If a buyer cannot distinguish a reliable used car worth £6000 from a faulty one worth £2000, a cautious offer of £4000 may discourage reliable-car sellers. Inspection or a credible warranty may improve the deal. It is not inevitable that only faulty cars remain; detailed information problems are a later A-Level extension.

Methods of government intervention

Indirect taxes

An **indirect tax** is charged on goods or services. A specific tax is a fixed amount per unit; its amount does not rise with the selling price. In the competitive market model, it creates a wedge between the buyer's price and the amount the

producer keeps. Keep these two prices separate.



tax: *S* shifts left · incidence split consumers/producers

The diagram's model is demand P=12−Q, supply P=2+Q, and a specific tax 3. Buyers pay 8.5 while producers keep 5.5. Both bear 1.5 of the tax relative to the original price 7.

Worked price wedge. Here P is pounds per unit and Q is units per period. Before tax, demand is P = 12 − Q and supply is P = 2 + Q:

$$P_D = P_S \quad \Rightarrow \quad 12 - Q = 2 + Q$$

$$Q_0 = (12 - 2)/2 = 5 \text{ units per period}$$

$$P_0 = 12 - Q_0 = 12 - 5 = £7 \text{ per unit}$$

A specific tax of £3 per unit changes the price buyers must pay for each supplied quantity to 5 + Q. The producer still keeps 2 + Q:

$$P_b = P_s + t \quad \Rightarrow \quad 12 - Q = 2 + Q + 3$$

$$Q_t = (12 - 5)/2 = 3.5 \text{ units per period}$$

$$P_b = 12 - Q_t = 12 - 3.5 = £8.5 \text{ per unit}$$

$$P_s = P_b - t = 8.5 - 3 = £5.5 \text{ per unit}$$

$$R = tQ_t = 3(3.5) = £10.5 \text{ per period}$$

The **incidence of a tax** is the distribution of the economic burden. Here buyers pay £1.5 more and producers keep £1.5 less per unit, so the burden is shared equally. The side legally sending the tax to government does not necessarily bear all of it.

Relative response matters. Holding supply conditions fixed, more **inelastic** demand allows a larger buyer-price increase than more **elastic** demand. When comparing who bears most, consider both demand and supply: the less responsive side bears more under the standard competitive model. Demand being inelastic alone does not prove that buyers bear most.

Counterexample: a different market. Demand is Qd = 120 − 2Pb and supply is Qs = 95 + 0.5Ps. The original price is £10 and quantity 100. At that point, absolute PED is 0.2 and PES is 0.05: demand is inelastic, but supply is even less responsive. A £2 tax gives Ps = Pb − 2:

$$Q_d = Q_s \quad \Rightarrow \quad 120 - 2P_b = 95 + 0.5(P_b - 2)$$

$$P_b = (120 - 94)/2.5 = £10.4 \text{ per unit}$$

$$P_s = P_b - 2 = 10.4 - 2 = £8.4 \text{ per unit}$$

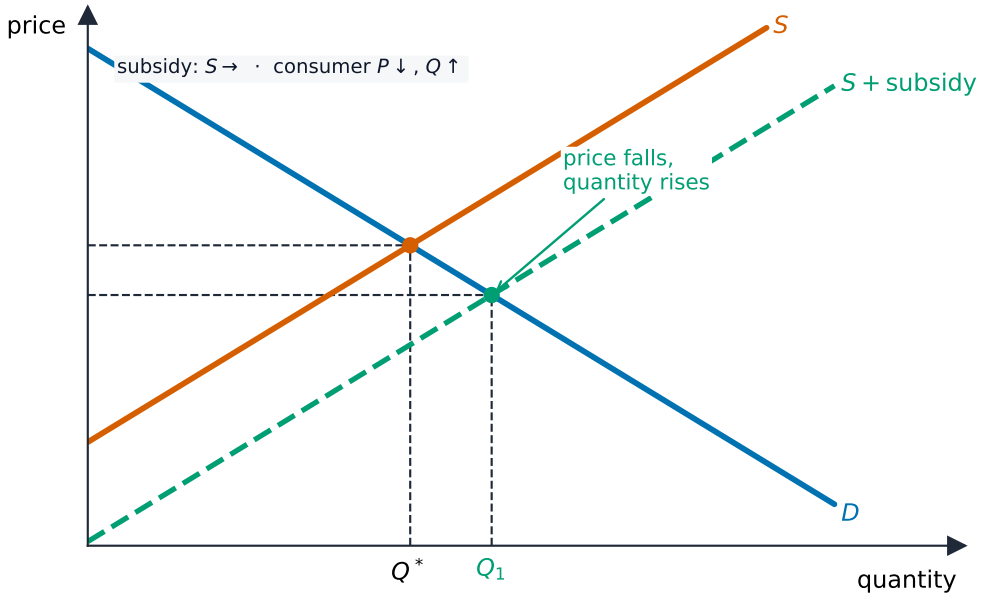
$$Q_t = 120 - 2P_b = 120 - 2(10.4) = 99.2 \text{ units per period}$$

Buyers bear 0.4/2 =20% of the tax; producers bear 1.6/2 =80%. This is a constructed counterexample, not an estimate for a real cigarette market.

Judge each policy aim separately. Raising revenue, reducing consumption and improving allocation are different outcomes. A small quantity fall can coexist with substantial revenue; it does not establish that the tax has no health benefit. For external costs, consider marginal harm and other conditions; not every tax moves output toward the social optimum. Whether its household burden is regressive also depends on spending and income patterns, not merely the name of the taxed good.

■ Subsidies

A **subsidy** is a government payment; this example is a specific payment to producers for each unit sold. It creates a wedge in the opposite direction: producers receive the buyer's payment plus the subsidy. Effects depend on responses and market conditions; a fixed grant is not automatically a per-unit supply shift.



This separate model uses demand P=12−Q, original supply P=2.5+Q and a producer subsidy 2.4 per unit. The new buyer price is 6.05; the producer receives 8.45 including the subsidy.

Worked producer subsidy. Before subsidy, equating 12 − Q and 2.5 + Q gives Q0 =4.75 and P0 =£7.25. With a subsidy £2.4 per unit, the buyer-facing supply price becomes 0.1 + Q:

$$P_D = P_S - s \quad \Rightarrow \quad 12 - Q = 2.5 + Q - 2.4$$

$$Q_{sub} = (12 - 0.1)/2 = 5.95 \text{ units per period}$$

$$P_b = 12 - Q_{sub} = 12 - 5.95 = £6.05 \text{ per unit}$$

$$P_s = P_b + s = 6.05 + 2.4 = £8.45 \text{ per unit}$$

$$G = sQ_{sub} = 2.4(5.95) = £14.28 \text{ per period}$$

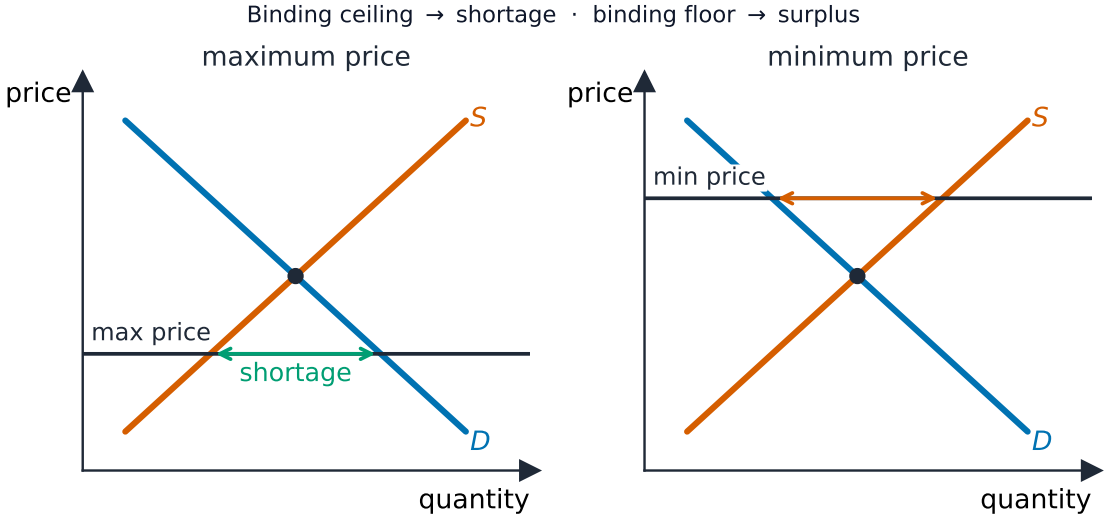
Buyer price falls by £1.2 and producer receipts rise by £1.2 per unit; neither side receives the whole benefit of the subsidy in this model. Government spending is not the same as the increase in producers' receipts relative to the old price.

Merit-good decision. A subsidy may make a training course cheaper, but low awareness, travel costs or a shortage of teachers may still limit attendance. Compare a production subsidy with information about benefits and direct provision of places.

Evaluate consumption achieved, who benefits and the opportunity cost of public funds. Do not conclude that any payment automatically corrects all underconsumption.

■ Maximum and minimum prices

A **maximum price** is a legal upper limit. It binds when below the relevant equilibrium; a higher ceiling may leave price and quantity unchanged. A **minimum price** is a lower limit, binding when above equilibrium. First test whether the control binds, then compare quantities demanded and supplied.



For the diagram's competitive market P=10−Q and P=Q, equilibrium is P5/Q5. A ceiling 3 creates a shortage 4; a floor 7 creates excess supply 4. The controls do not shift either curve.

Ceiling case. In this separate market, Qd =10 − P and Qs =P. At a ceiling £3:

$$Q_d = 10 - P_c = 10 - 3 = 7 \text{ units per period}$$

$$Q_s = P_c = 3 \text{ units per period}$$

$$\text{shortage} = Q_d - Q_s = 7 - 3 = 4 \text{ units per period}$$

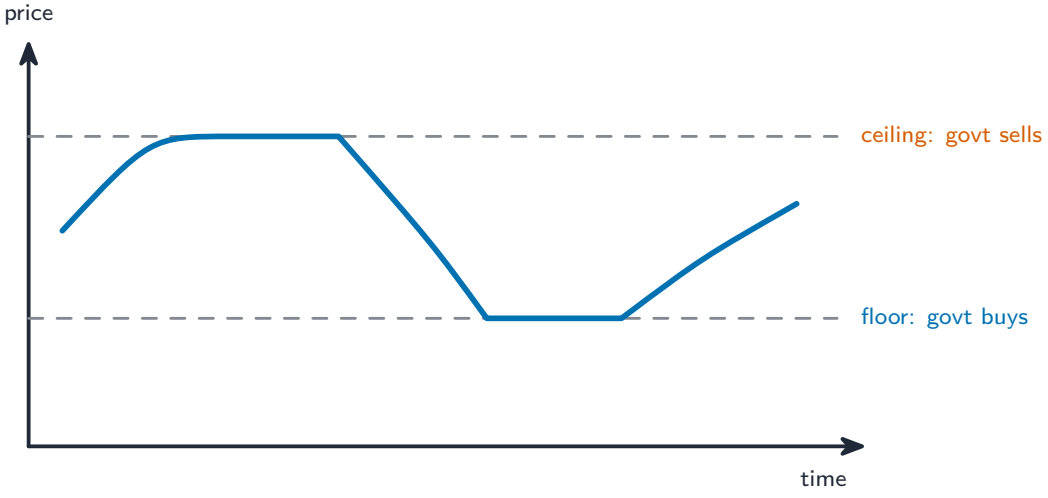
Without additional supply, only three units can be traded legally at that price. A **shortage** is not a promise that every buyer gets cheaper goods. Queues, allocation rules or an illegal resale market may result. A ceiling £6 is above equilibrium and does not bind.

Floor case. At a floor £7, Qs =7 but Qd =3, creating **excess supply** of four units. Without government purchases, private buyers take only three units. If government guarantees to buy the remaining four at the floor, its spending is 7(4)

=£28 per period and it must store or dispose of those units. A floor £4 does not bind. Changing the legal price is not the same as guaranteeing sales or household income.

Buffer stocks

A **buffer stock** scheme buys and stores a storable commodity when its price reaches a lower intervention price, then sells stock when price reaches the upper intervention price. Holding a band requires enough money and storage to buy, and enough usable stock to sell. It is not an automatic guarantee.



The band is a policy target. Purchases support the lower price; stock releases limit the higher price. The target can fail when funding or usable stocks run out.

Worked harvest decisions. Demand is P =10 − Q. Normal supply P =Q gives equilibrium P5/Q5. The government targets a band £4.5–£5.5. An opening store contains three units.

Harvest	Farm supply	Intervention price	Qd	Farm Qs
Bumper	P=Q−2	£4.5 floor	5.5	6.5
Poor	P=Q+2	£5.5 ceiling	4.5	3.5

With the bumper crop, free-market price would be £4. At the target floor, buy 6.5 −5.5 =1 unit for £4.5; stock rises 3→4. With the poor crop, free-market price would be £6. At the target ceiling, sell 4.5 −3.5 =1 unit for £5.5; stock falls 4→3. These numbers exclude storage costs and spoilage.

If three poor-harvest periods start with stock 3 and no purchases, one-unit releases empty the store. In a fourth poor period the scheme cannot supply the missing unit without another source. Perishable goods, changing long-run demand and persistent surpluses create further problems.

Price stability is not identical to income stability. Farmers' receipts here are 4.5(6.5)

=£29.25 after a bumper harvest, but 5.5(3.5) =£19.25 after a poor harvest: their own output differs. Selling stored stock does not give farmers the receipts from the current stock release. Compare buffer stocks with diversification, subsidies or other support when evaluating farm income.

Information and direct provision

State provision means that the **government** supplies or arranges a service. The service need not be free, and public funding does not by itself make it a public good. A free clinic still uses scarce appointments, staff and medicines. Direct provision can add capacity where a subsidy would only raise demand for fixed places; it needs finance, suitable staff and access for the intended users.

Choose an intervention. A district has 100 eligible children but only 60 childcare places. Forty additional places cost £4000 per period. A £1000 information campaign may help parents find existing vacancies, but it cannot remove this capacity shortfall if all 60 places are full. Compare the additional places with a different use of the same £4000, and check quality and location. Free childcare may enable parents to work; its total benefit is not measured only by the fee saved.

Later A-Level extension: regulation and permits

Regulation uses rules such as safety standards or pollution limits. **Tradable permits** allow firms to trade permission to emit under an enforced cap. Trading can move reductions towards firms able to cut pollution cheaply, but this depends on monitoring, enforcement and a functioning market; it does not guarantee minimum costs under all conditions. For example, if A can cut one unit for £10 and B for £30, making A cut meets a one-unit reduction target at lower resource cost. A trade can reward that cheaper reduction. The cap and compliance must still be checked.

Evaluating intervention and government failure

A policy being costly is not enough to show **government failure** . Compare its benefits and full costs with the outcome without the policy and with a relevant alternative. Failure means intervention makes resource allocation worse. Poor information, administration costs, unintended effects or decisions serving a narrow group can cause this. A binding ceiling can encourage an illegal **black market** , but that is a risk to analyse, not an automatic result of every price control. Formal government-failure analysis is developed in the later A-Level unit.

Illustrative comparison. A programme produces benefits worth £120 and uses resources costing £100, giving a net gain of £20. It is not a failure just because it costs £100. If another feasible programme gives benefits £140 for the same cost and the same relevant risks, its net gain is £40. Identify the objective, evidence and alternative before judging.

Income and wealth inequality

Income is a **flow** received over a period, such as wages, rent and interest. **Wealth** is a **stock** of assets owned at a point in time; net wealth subtracts debts. A person receiving £2000 each month does not therefore own £2000 of wealth. A house worth £150000 with a mortgage of £100000 contributes £50000 to net wealth. Savings can build wealth, and assets can produce income, but these are different measures.

■ Measuring and explaining inequality

The **Gini coefficient** summarises inequality on a scale from 0 (complete equality) to 1 (the complete-inequality limit). Specify whether the data measure income or wealth, and use a comparable population, period and definition. The AS syllabus requires interpretation, not calculation of the coefficient.

Worked interpretation. A fictional country's disposable-income Gini falls from 0.42 to 0.35. Its measured income distribution becomes more equal. This does not establish that every poor household receives more money, that average real income rises, that wealth becomes more equal, or that one policy caused the change. Government can use the measure to review redistribution, alongside household-group data, employment, prices and access to services. Two countries with the same Gini may have different average living standards and different distributions within income groups.

Income differences may reflect qualifications, experience, demand for occupations, hours worked, unemployment, discrimination or bargaining power. Wealth differences may reflect accumulated savings, inheritance, asset-price changes and debts. Causes interact: unequal access to education may affect future earnings, and inherited assets may generate rent. Do not rank one cause as universally most important without evidence.

■ Progressive taxes and transfer payments

A **progressive tax** takes a higher average **percentage** of higher incomes. Compare average rates, not just amounts paid. **Benefits**, also called **transfer payments**, are government payments without a current good or service received in return; eligibility differs between programmes.

Worked redistribution. Two households have annual incomes £10000 and £40000. Taxes of £1000 and £8000 give average rates 10% and 20%. If the lower-income household receives a £2000 transfer, their disposable incomes are:

$$Y_{d,L} = Y_L - T_L + B_L = 10000 - 1000 + 2000 = £11000 \text{ per year}$$

$$Y_{d,H} = Y_H - T_H + B_H = 40000 - 8000 + 0 = £32000 \text{ per year}$$

The income ratio falls from 4 to about 2.91. This is a two-household illustration, not a calculation of a national Gini or a claim that these taxes alone finance the entire

government. Equal £1000 taxes instead take 10% and 2.5% of the original incomes; paying more pounds is not the definition of progressivity.

Minimum wages and access to essential services

A minimum wage is a legal floor on hourly pay. In an illustrative competitive labour market, demand is Ld =100 −5w and supply is Ls =5w. The original hourly wage is £10 and employment 50 workers. A binding £12 floor gives demand 40 and supply 60: 40 jobs and excess labour supply of 20 under this model. Workers who keep the same hours earn more, but excluded workers do not receive that wage. Household effects also depend on hours, other earners, prices and benefits. A floor £8 does not bind here. Other labour-market structures and their full analysis are developed later; the competitive prediction is not a universal empirical law.

Free or subsidised essential services can raise real living standards without increasing cash income. A £500 saving on childcare can matter to a household even if its measured cash income is unchanged. Funding, quality, waiting times and location determine whether poorer households can actually use the service. Education and childcare access may also improve future earnings; redistribution need not always reduce efficiency.

Income taxes compared with inheritance and capital taxes

Inheritance taxes target transfers of assets after a death; capital taxes target specified assets, ownership or gains, depending on the policy. They can address accumulated wealth that an income tax misses. Accurate valuation, debt, exemptions, collection and avoidance matter. A valuable house may produce little cash for paying a tax, while assets with reliable ownership records may be easier to assess than hidden business holdings.

Policy judgement. Regular recorded wages can make income taxation and transfers easier to administer than taxes on hard-to-value wealth. This is a conditional argument, not a rule that wealth can never be redistributed. Land records or estate information can assist wealth taxation; informal earnings can make income taxation difficult too. Assess which inequality is targeted, administrative capacity, avoidance and the effects on work, investment and opportunity. This **redistribution** can involve a **trade-off** when marginal tax and benefit rules weaken an **incentive** to earn more, but the strength of that response needs evidence.



Different housing conditions may prompt investigation; this photograph alone cannot measure incomes, wealth, poverty or a Gini coefficient.

Exam tips

- Identify the actual problem: public-good financing, misjudged benefit or harm, a price objective, or income/wealth inequality.
- Explain the mechanism with a relevant example, diagram or calculation. Distinguish prices, quantities, fiscal costs and household outcomes.
- Evaluate against the stated aim and a feasible alternative. Give a reasoned conclusion using conditions, not a fixed list of advantages and disadvantages.
- In tax incidence compare both demand and supply responses. For Gini data explain what the measure can and cannot establish; no Gini calculation is required.
- The externality diagrams, monopoly discussion and pollution permits in this handout are later A-Level previews. Study the AS core requirements first.

Vocabulary card meanings

private cost: A cost paid by the person or firm that causes it. external benefit: A gain received by a third party who is outside the deal between buyer and seller. buffer stock: A store of a commodity kept to steady its price. black market: Trade that breaks a legal rule on price or quantity. wealth: The value of what a person owns after subtracting what that person owes, measured at one moment. stock: A quantity measured at one moment, not over a period of time. benefits: Payments from the state to people who qualify for help.

The macroeconomy

A-Level Economics

Macroeconomics studies the economy as a whole: output, jobs and prices. A bakery can sell more bread because it produces more, because prices rise, or both. National accounts help us separate these changes before judging economic performance.

National income statistics

National income describes income generated by production during a period. GDP, GNI and NNI are related measures, but answer different questions. State the period, currency and price basis when comparing them.



GDP measures domestic production; GNI measures residents' primary income.

■ Count final output or value added

Gross domestic product (GDP) is the value of final goods and services produced within an economy during a period. It includes government services, not just goods sold by private firms. Intermediate inputs are used to produce other output; do not count them again alongside that final output.

Value added is output value minus the value of intermediate inputs bought from other producers.

Worked example – one domestic bread chain. All amounts are £ per batch. Assume no other bought inputs or product taxes.

Producer	Sales	Bought inputs	Value added
Farmer	20	0	20
Miller	35	20	15
Baker	60	35	25

The farmer sells grain, the miller sells flour and the baker sells the final bread to households. Adding sales gives £115, but counts the grain and flour more than once. Adding value added gives £60, the same as final bread sales.

■ Domestic, national, gross and net

Gross national income (GNI) is residents' total primary income. Start with GDP, add primary income residents receive from abroad and subtract primary income paid to non-residents. Residency is different from citizenship. Export receipts are payments for output, not primary income from abroad.

$$\text{GNI} = \text{GDP} + \text{primary income received} - \text{primary income paid}$$

Worked example – annual accounts. GDP is £800 billion/year, primary income received from abroad is £50 billion/year and income paid abroad is £70 billion/year, all at the same price basis.

$$\text{GNI} = \text{GDP} + \text{income received} - \text{income paid} = 800 + 50 - 70 = 780 \text{ (£ billion/year)}$$

A foreign-owned factory's domestic output counts in GDP. Income it pays to non-residents enters the GNI adjustment; we do not remove its entire output simply because its owners live abroad.

Capital depreciation means consumption of fixed capital through wear and obsolescence. **Net national income** (NNI) deducts this from GNI. If capital depreciation is £60 billion/year:

$$\text{NNI} = \text{GNI} - \text{capital depreciation} = 780 - 60 = 720 \text{ (£ billion/year)}$$

Gross/net adjusts for capital used up. It does not remove inflation.

■ Market prices and basic prices

Market prices include taxes on products less subsidies on products. **Basic prices** remove this net product-tax component. For the syllabus conversion:

$$\text{GDP at basic prices} = \text{GDP at market prices} - \text{taxes on products} + \text{subsidies on products}$$

Worked example – separate accounts. GDP at market prices is £500 billion/year, product taxes £80 billion/year and product subsidies £20 billion/year.

$$\text{GDP}_b = \text{GDP}_m - \text{product taxes} + \text{product subsidies} = 500 - 80 + 20 = 440 \text{ (£ billion/year)}$$

Reverse the signs to check the original market-price value:

$$\text{GDP}_m = \text{GDP}_b + \text{product taxes} - \text{product subsidies} = 440 + 80 - 20 = 500 \text{ (£ billion/year)}$$

Do not subtract personal income tax or capital depreciation in this price-basis step.

■ Nominal, real and per capita

Nominal values use current prices. **Real** values remove price changes so that changes in the amount of output can be compared. Nominal figures remain useful for current money receipts and payments.

Worked example – separate two-year series. Nominal GDP is £800 billion/year in Year 0 and £924 billion/year in Year 1. The given GDP price index rises from 100 to 110, with Year 0 as the base. Use a GDP price index, rather than assuming a consumer price index measures all GDP prices.

$$\text{Real GDP} = \frac{\text{nominal GDP}}{\text{GDP price index}} \times 100$$

$$\text{Real GDP}_1 = \frac{\text{nominal GDP}_1}{\text{GDP price index}_1} \times 100 = \frac{924}{110} \times 100 = 840 \text{ (£ billion/year)}$$

$$\text{Real growth} = \frac{\text{real GDP}_1 - \text{real GDP}_0}{\text{real GDP}_0} \times 100\% = \frac{840 - 800}{800} \times 100\% = 5\%$$

Per capita means per person. If population rises from 40 million to 42 million in this example:

$$\text{Real GDP per capita}_0 = \frac{\text{real GDP}_0}{\text{population}_0} = \frac{800 \text{ billion £/year}}{40 \text{ million people}} = \text{£20,000/person/year}$$

$$\text{Real GDP per capita}_1 = \frac{\text{real GDP}_1}{\text{population}_1} = \frac{840 \text{ billion £/year}}{42 \text{ million people}} = \text{£20,000/person/year}$$

Output and population both rise 5%, leaving real output per person unchanged. The average does not show who receives income.

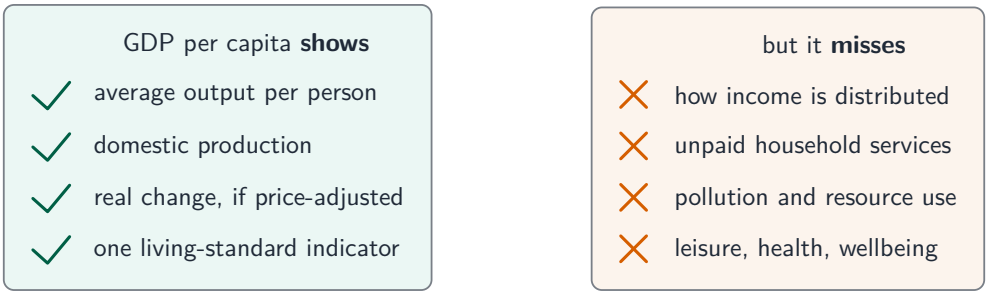
A lower positive growth rate does not mean falling output. If a real output index starts at 100, rises 4% and then 2%, its levels are 104 and 106.08. Growth slows by 2 percentage points but output rises in both years. Distinguish a forecast from an observed outcome.

Uses and limitations

National income statistics help compare output over time and resources available for consumption and public services. For comparisons across countries, check prices, currency conversion, purchasing power, population and data quality.

The **standard of living** is wider than output per person. Consider **distribution**, leisure, health, access to services and pollution. For example, two economies can have the same average but give the poorest fifth 10% and 3% of income respectively. The average alone cannot rank their living conditions.

Most unpaid household services, such as cleaning your own home, are outside GDP; paying someone to provide the service can count. The **informal economy** includes activity outside or insufficiently covered by formal arrangements. Production within the GDP boundary should be included in principle, but it can be difficult to measure fully. Do not claim that all informal production is excluded because it is untaxed.



Use real GDP per person as one indicator, alongside evidence about living conditions and measurement quality.

The circular flow of income

The **circular flow of income** links production, income and expenditure. **Households** supply factor services to **firms** and receive wages, rent, interest and profit. Households buy goods and services from firms.

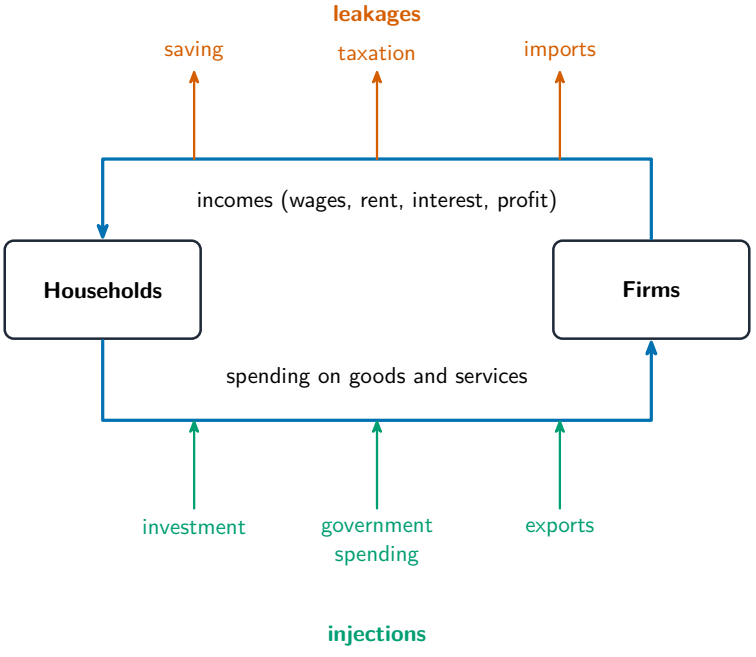
Transaction	Real flow	Money flow
Employment	Labour to firms	Wages to households
Bread purchase	Bread to households	Spending to firms

In each transaction the real flow and its payment go in opposite directions. Saving means households need not spend all current income on consumption.

Leakages and injections

Leakages are **saving** (S), **taxation** (T) and **imports** (M). **Injections** are **investment** (I), **government spending** on goods and services (G), and **exports** (X).

A newly produced factory machine is investment; buying an existing share is an asset transaction, not new capital production. Government purchases of teaching services count in G. A pension is a transfer payment, rather than payment for current output; the recipient's later purchases may enter consumption. Imported shoes enter household consumption and imports: subtracting M prevents foreign production from being counted as domestic GDP.



Saving, taxes and imports leave the domestic spending loop; investment, government purchases and exports enter it.

A **closed economy** has no foreign trade, so X and M are zero. An **open economy** includes international transactions. Government can be present in either.

Equilibrium and disequilibrium

Equilibrium in this simple model requires planned injections to equal planned leakages at the current income level. Actual expenditure accounts include unplanned inventory changes; do not mistake an accounting identity for proof that plans are in balance.

Worked example – annual planned flows at fixed prices. Income Y is £300 billion/year; S = 35, T = 25, M = 30, I = 30, G = 40 and X = 20, all in £ billion/year.

Assume no transfer payments. Find consumption C:

$$C = Y - S - T = 300 - 35 - 25 = 240 \text{ (£ billion/year)}$$

$$\text{Planned injections} = I + G + X = 30 + 40 + 20 = 90 \text{ (£ billion/year)}$$

$$\text{Planned leakages} = S + T + M = 35 + 25 + 30 = 90 \text{ (£ billion/year)}$$

$$\text{Planned expenditure} = C + I + G + X - M = 240 + 30 + 40 + 20 - 30 = 300 \text{ (£ billion/year)}$$

Planned expenditure equals current output, so there is no unplanned inventory change.

Change the plan. Investment rises to 40 while current output and other planned flows initially stay fixed.

$$\text{Planned expenditure} = C + I + G + X - M = 240 + 40 + 40 + 20 - 30 = 310 \text{ (£ billion/year)}$$

$$\text{Unplanned inventory change} = Y - \text{planned expenditure} = 300 - 310 = -10 \text{ (£ billion/year)}$$

Unexpected inventory falls give firms a reason to raise production and employment if spare capacity allows. Near capacity, extra demand may mainly raise prices. As income changes, saving, taxes and imports may also change. An initial excess of injections does not guarantee a particular rise in real output. If planned spending falls below output, inventories rise unexpectedly and firms may reduce production. No multiplier calculation is needed here.

Check yourself

New annual data, all £ billion at the same price basis: GDP at market prices 600; product taxes 90; product subsidies 10; primary income received 30 and paid 50; capital depreciation 40. Find GDP at basic prices and NNI at market prices. Explain why neither answer alone establishes real growth.

Worked answer. The price-basis adjustment is:

$$\text{GDP}_b = \text{GDP}_m - \text{product taxes} + \text{product subsidies} = 600 - 90 + 10 = 520 \text{ (£ billion/year)}$$

The resident-income and capital-use adjustments are:

$$\text{NNI}_m = \text{GDP}_m + \text{income received} - \text{income paid} - \text{capital depreciation} = 600 + 30 - 50 - 40 = 540 \text{ (£ billion/year)}$$

Real growth needs comparable real output for at least two periods. These are different measures for one period.

Aggregate demand and aggregate supply

Aggregate demand

Aggregate demand (AD) is planned spending on domestic goods and services at each price level. Its components are household **consumption** C, spending on new capital I, government purchases of goods and services G, and **net exports** X − M.

$$AD = C + I + G + (X - M)$$

Worked example – stated price level. Planned annual spending is C = 150, I = 30, G = 40, X = 25 and M = 35, all £ billion/year.

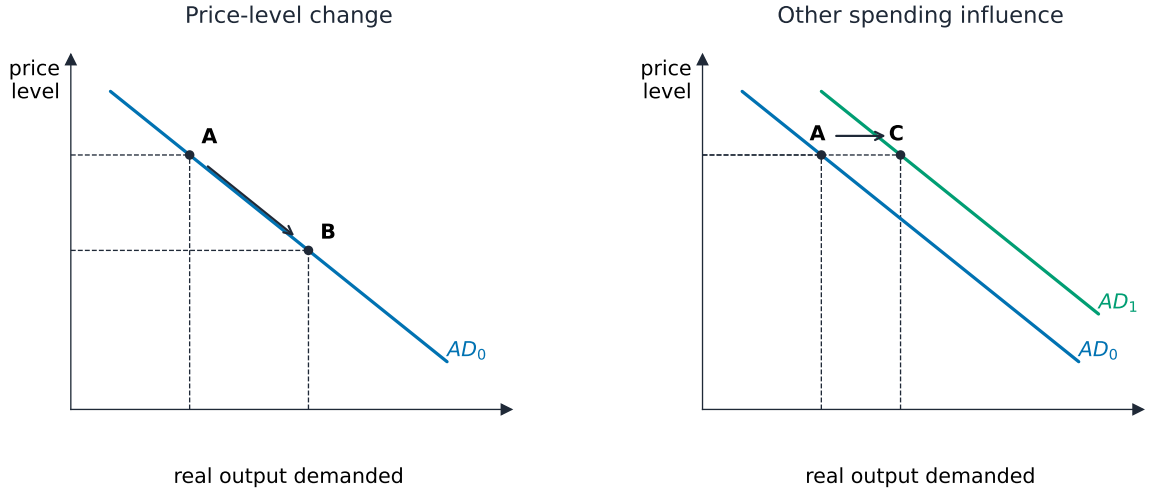
$$AD = C + I + G + (X - M) = 150 + 30 + 40 + (25 - 35) = 210 \text{ (£ billion/year)}$$

Net exports are negative: imports can exceed exports. Imported products can appear in C, I or G, so subtract M to remove foreign production. Buying an existing share is not spending on newly produced capital, and a pension is not a government purchase of current output.

A movement along AD and a shift

Hold other influences constant. A lower domestic price level can raise the purchasing power of money balances and make domestic goods cheaper relative to foreign goods. Real output demanded rises along the existing AD curve. This helps explain its downward slope.

By contrast, higher household confidence can increase consumption at each price level, shifting AD right. A change in a non-price influence changes the relationship, rather than moving along it.



A to B moves along AD after a price-level change. A to C shifts demand at the same price level.

Explain the decisions. A tax cut raises disposable income and may raise C, but households may save or repay debt; some purchases may be imports. Higher expected sales may encourage firms to buy new machinery, raising I, but spare existing capacity, expensive credit or uncertainty can weaken the response. Higher government purchases directly raise G. Stronger foreign demand may raise X. Explain the component and mechanism before drawing an AD shift; do not assume every change in income or borrowing causes the same spending response.

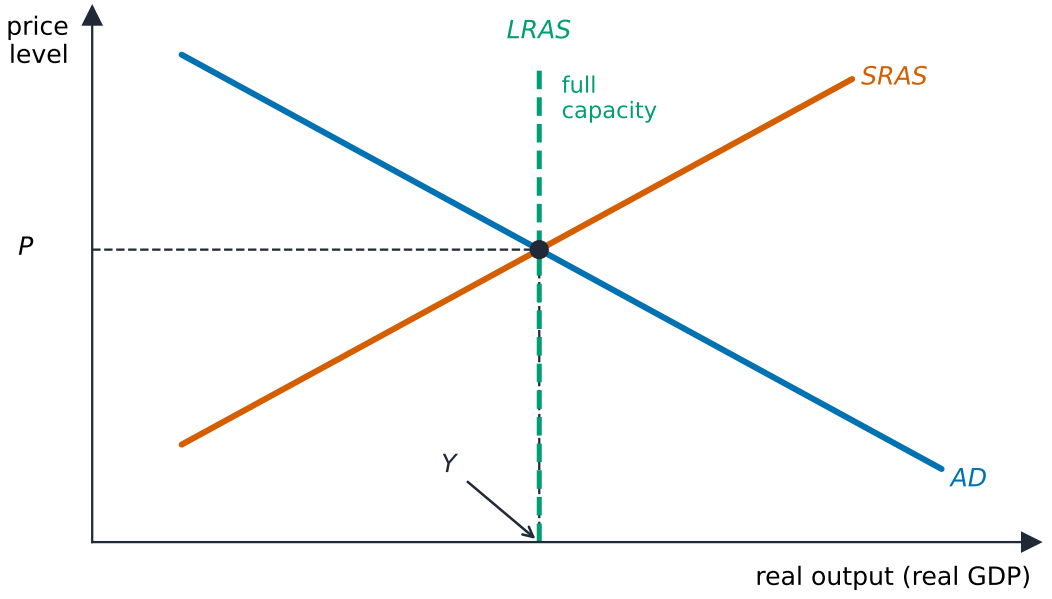
Aggregate supply

Aggregate supply (AS) relates the price level to real output supplied. **Short-run aggregate supply** (SRAS) slopes upward when some input costs adjust slowly. If output prices rise while wage rates and input prices initially stay fixed, producing extra output becomes more attractive: move up the same SRAS.

Higher imported fuel costs reduce profitable supply at each price level and shift SRAS left/up. Better **productivity** can reduce unit costs and shift SRAS right. Separate a change in the price level from a change in a supply determinant.

Macroeconomic equilibrium

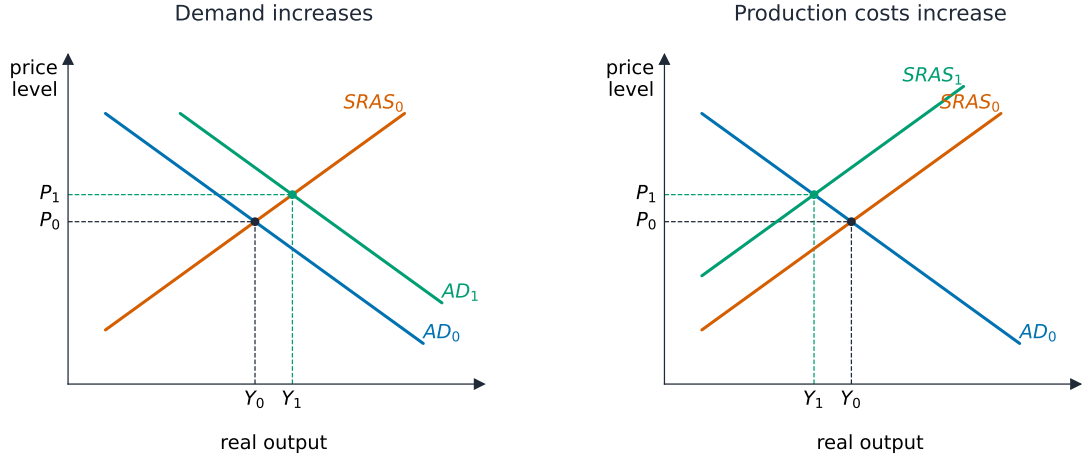
Macroeconomic equilibrium is the relevant AD/AS intersection. It determines the **price level** and **real output** in that model.



Here the current intersection happens to lie at capacity. An AD/SRAS intersection need not always lie on LRAS.

Compare changes in demand and supply

The paired diagrams below use a separate illustrative short-run model with upward-sloping SRAS and room for output to change. Each starts at the same initial intersection.



Demand growth raises both the price level and output; higher costs raise prices but reduce output, other influences unchanged.

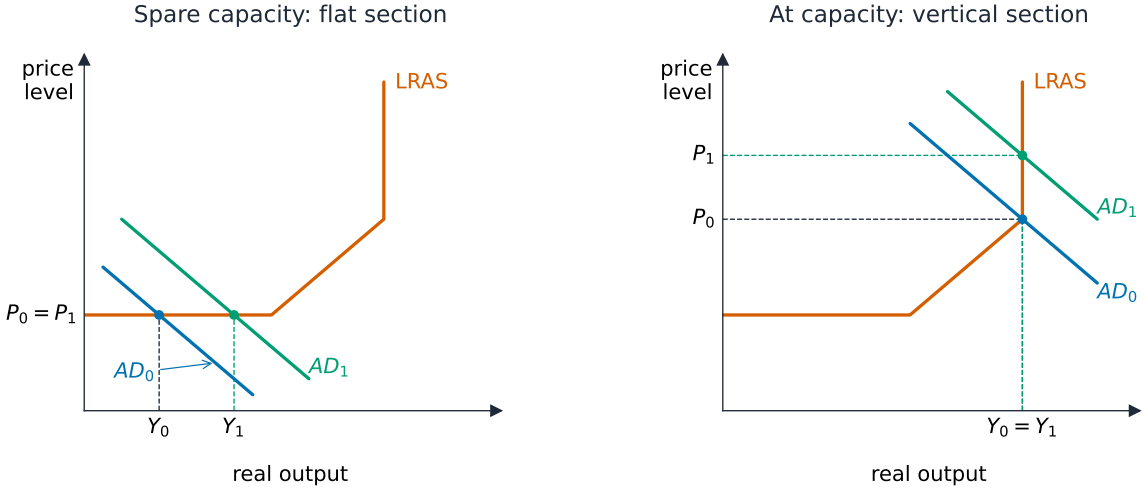
- When AD shifts right along upward-sloping SRAS, real output and the price level rise. Firms may need more workers; the employment response also depends on productivity and hours worked.
- When SRAS shifts left with AD unchanged, the price level rises while real output falls. Firms may reduce employment as they reduce production. This is a different mechanism from higher spending.

A shift showing a higher price level illustrates inflation pressure between periods. Persistent inflation means prices keep rising over time; one shock alone does not establish a permanently continuing inflation rate.

Capacity and the long run

Long-run aggregate supply (LRAS) depends on **productive capacity**: resources, skills, technology and productivity. More reliable electricity can reduce factory outages and expand sustainable production. Extra household spending initially shifts AD. Some spending, such as infrastructure construction, can raise demand now and improve supply later.

The syllabus permits vertical LRAS or a three-section representation: a highly elastic/flat section with spare capacity, an upward section as constraints increase, and a vertical section at capacity.



With spare capacity, output can rise without a higher price level. At capacity, extra demand can raise prices without increasing real output.

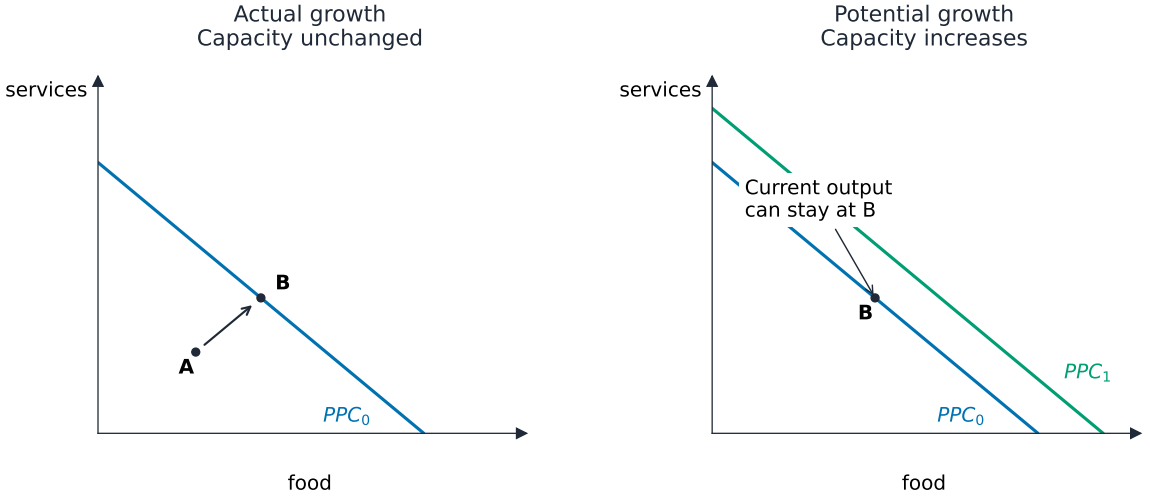
In the upward section, both output and prices can rise. Identify the model, the capacity position and the time period before concluding that every AD or AS shift changes both variables. Extra capacity does not guarantee enough demand to use it. These conditions explain why an increase in AD does not always lead to inflation.

Economic growth

Economic growth is an increase in real output over time. **Actual growth** is the observed increase. **Potential growth** expands what the economy can sustainably produce.

Compare two changes. Restarting idle factories raises current output using existing capacity. Better machinery can expand capacity and shift LRAS right, even before firms receive more orders. Actual growth can occur without new capacity; potential growth can occur before actual output rises.

A **production possibility curve** (PPC) shows feasible output combinations using available resources and technology.



Left: A to B uses existing capacity. Right: the PPC expands, but current output can initially remain at B.

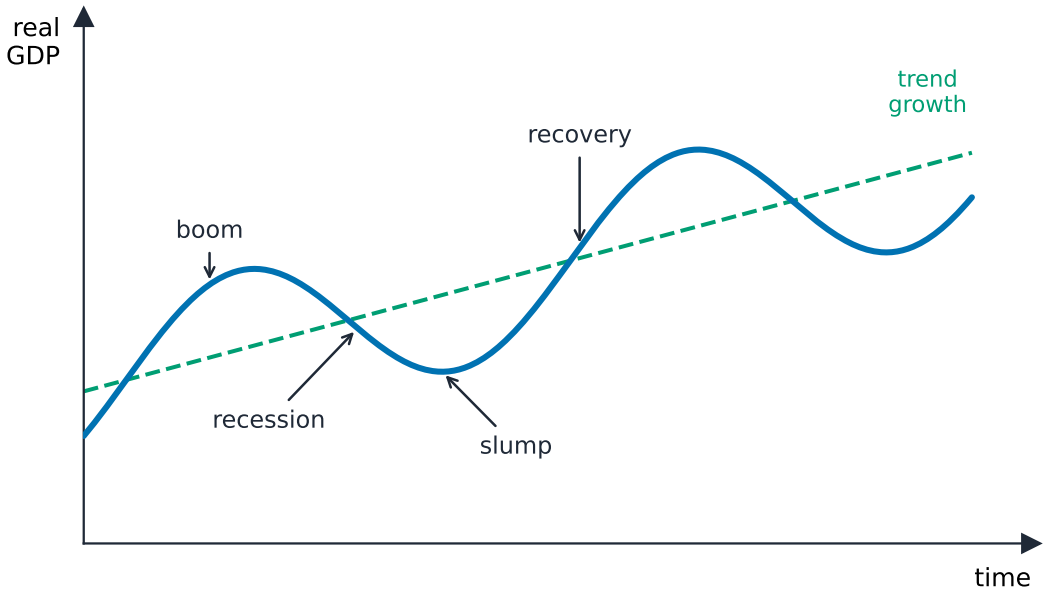
Causes of growth

More working-age residents can increase labour input; training can reduce defects and raise useful output per worker. Investment can improve capital, while technology and infrastructure can increase productivity. These mechanisms require suitable resources and demand: an unused machine does not automatically increase actual output.

For real output per person to rise, real output must grow faster than population. The earlier example shows why 5% real GDP growth alongside 5% population growth leaves the average unchanged. Distribution also affects who gains.

The business cycle

Real output does not grow smoothly. The **business cycle** includes **boom**, **recession**, **slump** and **recovery**. A recession involves falling real output; a recovery involves rising output after a downturn. Definitions using a particular quarterly rule require suitable quarterly data.



Read the curve's slope to judge output growth; being below the trend does not itself mean output is falling.

Worked example – separate annual output series. The real-output index is 104 in Year 1, 102 in Year 2 and 103 in Year 3.

$$g_t = \frac{Y_t - Y_{t-1}}{Y_{t-1}} \times 100\%$$

$$g_2 = \frac{Y_2 - Y_1}{Y_1} \times 100\% = \frac{102 - 104}{104} \times 100\% \approx -1.92\%$$

$$g_3 = \frac{Y_3 - Y_2}{Y_2} \times 100\% = \frac{103 - 102}{102} \times 100\% \approx 0.98\%$$

Output falls and then rises, but 103 is still below 104. Recovery does not mean the previous output level has already been regained. A slower positive growth rate is also different from negative growth.

■ Costs and benefits of growth

Higher output can support incomes, jobs, tax receipts and public services. But examine employment, real output per person and distribution before assuming that everyone benefits equally or poverty falls.

Compare two illustrative paths: A reopens idle low-pollution factories; B produces more through longer working hours and more polluting activity. Both can raise GDP, but B may lose leisure and face health costs or resource depletion. Growth caused by extra demand near capacity can create inflation pressure; growth expanding supply can ease pressure. Judge the mechanism, scale, sustainability and who gains or loses.

Unemployment

unemployment means having no job while available for work and actively seeking it. The **labour force** is employed people plus unemployed people. People neither working nor actively seeking work are outside it; they are economically inactive. Classify students and retired people by their actual work and search status, not their label alone.

■ Measuring unemployment

The **claimant count** counts people claiming specified unemployment benefits. Administrative records are relatively cheap to collect, but miss non-claimants and ineligible jobseekers. Changes in eligibility can change the count without a change in jobs.

The **labour force survey** asks a representative sample about work, availability and active search. A common international definition helps comparisons, but sampling error, inaccurate replies and differences in implementation affect estimates. Discouraged people who stop searching are outside measured unemployment. A person working fewer hours than they want can still be classified as employed.

> **Worked example: choose the denominator.** An illustrative population has 1000 people: 540 employed, 60 without jobs but available and seeking work, 150 inactive working-age people and 250 children. Let employed people be E and unemployed people be U.

$$\text{labour force} = E + U = 540 + 60 = 600 \text{ people.}$$

The unemployment rate measures the unemployed share of the labour force, not the whole population:

$$\text{unemployment rate} = \frac{U}{E + U} \times 100\%.$$

$$\text{unemployment rate} = \frac{60}{540 + 60} \times 100\% = 10\%.$$

Dividing by 1000 would give 6%, the unemployed share of the total population, which is a different measure.

> **Worked example: a rate can fall without new jobs.** Forty of these unemployed people stop seeking work; nobody gains or loses a job. Unemployment becomes 20 and the labour force becomes 560.

$$\text{unemployment rate} = \frac{U}{E + U} \times 100\%.$$

$$\text{unemployment rate} = \frac{20}{540 + 20} \times 100\% \approx 3.57\%.$$

Employment stays at 540. Read employment and participation alongside the unemployment rate before concluding that jobs have improved.

Types and causes

- **frictional unemployment** : matching a worker to a vacancy takes time, for example while searching between jobs. It is often brief, but its duration is not fixed.
- **structural unemployment** : changes in production and demand leave workers' skills or locations mismatched with available jobs, for example after a local industry declines.
- **cyclical unemployment** : weak AD reduces sales and firms' demand for labour during a downturn. Its size and duration depend on the downturn and recovery.
- **seasonal unemployment** : predictable seasonal changes remove jobs at certain times, for example winter layoffs at a summer resort.
- **technological unemployment** : machines or software replace particular labour tasks. It overlaps with structural unemployment when displaced workers lack the skills needed for new jobs. Automation can also create jobs and reduce costs elsewhere; the displaced workers do not necessarily gain those jobs immediately. Do not count overlapping causes twice.

Consequences and comparison

Lost earnings can reduce household spending and living standards. Long spells without work can harm skills and health; savings, benefits and access to another job affect the harm. Unused willing labour means forgone output, and falling household spending can weaken firms' sales further. Production may lie inside the PPC when usable resources are idle, although jobseekers cannot necessarily fill every vacancy immediately.

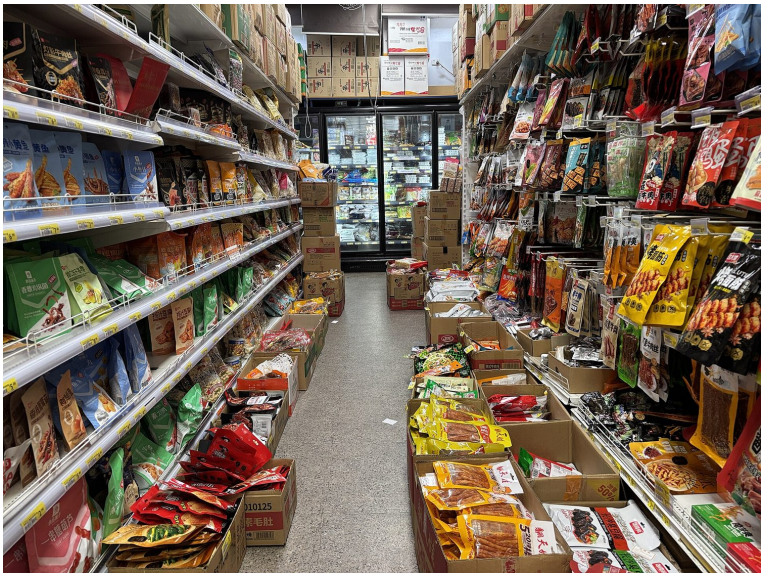
Lower taxable earnings reduce government receipts; eligible benefit claims raise spending. The effect depends on tax and benefit rules, numbers affected and duration.

Worked comparison. In illustrative case A, 2000 workers lose jobs after a local industry closes and nearby vacancies need different skills. In B, 20000 workers lose jobs across industries during a demand downturn. A may cause persistent regional harm if retraining and moving are difficult. B affects ten times as many people and may cause a larger immediate loss of income and output. Neither type is always worse: compare numbers, duration, regional concentration, lost income and routes back to work. Without these data, a definite ranking is unjustified. Stronger AD helps weak sales; training helps a mismatch, but neither guarantees a suitable job. Unit 5 develops the policy analysis.

Price stability

Inflation, deflation and disinflation

inflation is a sustained rise in the general price level. Money's **purchasing power** falls: a given sum buys less. **deflation** is a sustained fall in the general price level. **disinflation** means inflation slows while remaining positive: prices still rise, more slowly. A fall in one product's price does not establish economy-wide deflation.



A CPI follows prices across a household consumption basket, not just one product.

Building and interpreting a CPI

The **consumer price index** (CPI) tracks prices of a household consumption basket against a base period. A **weight** reflects a category's share of household spending, not the number of products. In this simplified calculation, category price indices use the same base year and weights sum to 1:

$$\text{CPI} = \sum_i w_i P_i.$$

> **Worked example: a two-category basket.** Food has weight 0.60 and price index 120; transport has weight 0.40 and index 110.

$$\text{CPI} = (0.60 \times 120) + (0.40 \times 110) = 116.$$

The simple average, 115, incorrectly gives equal weight to unequal spending shares. With a base-year index of 100, this basket costs 16% more than in the base year.

Next year, food's index is 126 and transport's is 121. Keep the same weights:

$$\mathrm{CPI}_t = \sum_i w_i P_{i,t}.$$

$$\mathrm{CPI}_t = (0.60 \times 126) + (0.40 \times 121) = 124.$$

The inflation rate is the percentage change from the previous index:

$$\pi_t = \frac{\mathrm{CPI}_t - \mathrm{CPI}_{t-1}}{\mathrm{CPI}_{t-1}} \times 100\%.$$

$$\pi_t = \frac{124 - 116}{116} \times 100\% \approx 6.90\%.$$

The index rose 8 points, but prices rose about 6.90%, not 8%.

Worked example: disinflation. A separate annual CPI series is 100, 104, then 107.

$$\pi_1 = \frac{\mathrm{CPI}_1 - \mathrm{CPI}_0}{\mathrm{CPI}_0} \times 100\% = \frac{104 - 100}{100} \times 100\% = 4\%.$$

$$\pi_2 = \frac{\mathrm{CPI}_2 - \mathrm{CPI}_1}{\mathrm{CPI}_1} \times 100\% = \frac{107 - 104}{104} \times 100\% \approx 2.88\%.$$

The rate falls, but the price level still rises: disinflation, not deflation.

■ Difficulties in measurement

An average basket does not represent every household. Using food index 120 and transport index 110, a household spending 80% on food and 20% on transport has a different cost index:

$$P_H = w_f P_f + w_t P_t.$$

$$P_H = (0.80 \times 120) + (0.20 \times 110) = 118.$$

Its basket costs 18% more than in the common base year, compared with 16% for the official example. Basket updates can lag changes in buying habits. Quality changes, new products and representative price sampling also make like-for-like measurement difficult.

■ Nominal and real pay

Worked example. Weekly nominal pay rises from £500 to £520 while the CPI rises from 100 to 104. Express real pay in base-year pounds:

$$\text{real pay} = \frac{\text{nominal pay}}{\mathrm{CPI}} \times 100.$$

$$\text{real pay}_1 = \frac{£520/\text{week}}{104} \times 100 = £500/\text{week}.$$

Initial real pay was also £500 per week. The pay rise buys the same basket. Wider living standards also depend on working hours, other income, household needs and public services.

Practice with solution. Pay rises from £500 to £550; CPI rises from 100 to 105. Find real pay and its percentage change before reading on.

$$W_1^R = \frac{W_1^N}{\text{CPI}_1} \times 100 = \frac{£550/\text{week}}{105} \times 100 \approx £523.81/\text{week}.$$

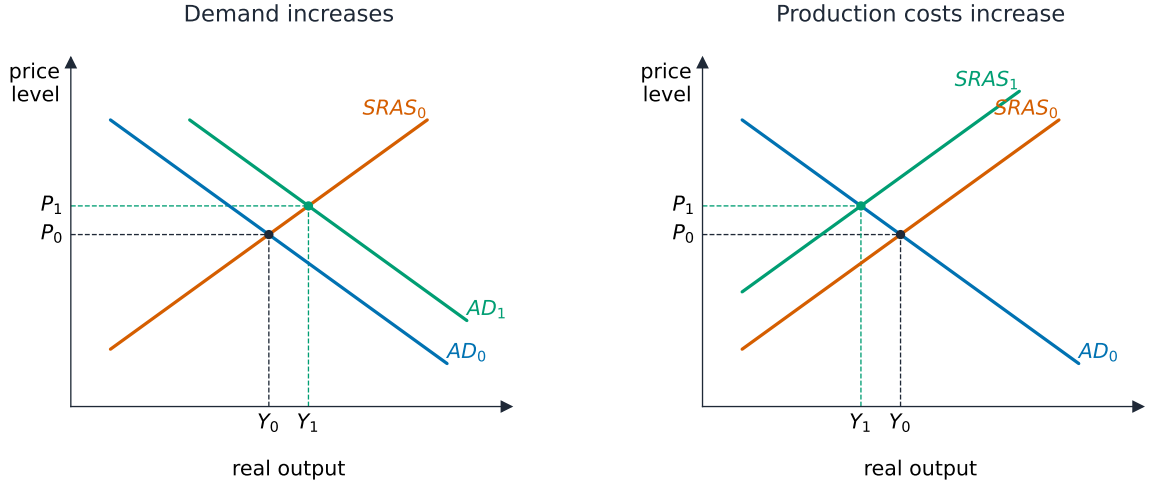
$$g_R = \frac{W_1^R - W_0^R}{W_0^R} \times 100\% = \frac{523.81 - 500}{500} \times 100\% \approx 4.76\%.$$

Subtracting nominal and price growth rates gives an approximation ($10\% - 5\% = 5\%$), not the exact change.

Causes of inflation

demand-pull inflation comes from stronger planned spending shifting AD right. Price pressure depends on spare capacity and the supply response; an AD increase need not cause a large price rise. Investment can raise AD now and productive capacity later.

cost-push inflation comes from higher input costs shifting SRAS left. With AD unchanged and upward-sloping SRAS, the new equilibrium has a higher price level and lower real output.



These separate short-run models hold the other curve unchanged. Spare capacity and later supply changes affect the outcome.

A one-off fuel-cost increase can raise the price level without causing repeated annual price rises. Continuing inflation requires continuing pressure or further adjustments. State the time period and AS conditions rather than treating every increase in spending or costs as continuing inflation.

■ Consequences of inflation

For consumers, income that fails to keep pace loses purchasing power. The effect varies with spending patterns, wage adjustment and savings. High, unpredictable inflation can make planning harder and weaken confidence.

> **Worked example: fixed nominal debt.** A debt stays at £1000 while the price index rises from 100 to 120.

$$D^R = \frac{D^N}{P} \times 100.$$

$$D^R = \frac{£1000}{120} \times 100 \approx £833.33.$$

Its real value falls from £1000 in base-year pounds. Unexpected inflation can favour a fixed-rate borrower over the lender. Indexed debts or higher interest charges can offset this; repayment ability depends on income. Savers' real returns depend on the interest earned relative to inflation. Borrowers do not always gain and savers do not always lose.

For firms, higher prices do not guarantee higher profit. Demand-led inflation may support sales, whereas cost-push inflation may squeeze margins. Export competitiveness depends on foreign prices and exchange rates as well as domestic inflation.

> **Worked example: price, cost and sales.** An illustrative firm has no other costs. Price starts at £10 per unit, unit cost £8 and sales 100 units. Later price is £12, unit cost £11 and sales 80 units.

$$\text{profit} = (p - c)Q.$$

$$\text{profit}_0 = (£10 - £8)/\text{unit} \times 100 \text{ units} = £200.$$

$$\text{profit}_1 = (£12 - £11)/\text{unit} \times 80 \text{ units} = £80.$$

Selling prices rise but profit falls because margins and sales fall. High inflation also makes frequent repricing costly; uncertain costs and demand can discourage investment and make price signals harder to interpret.

Assessment practice. Is high inflation always more damaging to consumers than to firms? Compare a fixed-income household buying essentials with a firm using imported inputs that cannot quickly raise selling prices. Explain each mechanism, then

judge using income adjustment, debt terms, spending patterns, market power, input exposure and duration. Keep the answer about **high** inflation if that is the question's context. Neither group is always harmed more.

■ Why deflation's cause matters

A collapse in AD can lower both prices and output, increasing unemployment. Expectations of further price falls may delay some purchases, especially durable goods, weakening demand further. Fixed nominal debts become more burdensome in real terms. But productivity improvements can lower prices while raising output. Falling prices alone do not prove a demand collapse: cause, expectations and debt exposure determine the risks.

Exam practice and checks

1. Nominal GDP rises 8% while the GDP price index rises 10%. Does real output rise? Explain rather than subtracting rates as an exact calculation.
2. The unemployment rate falls with employment unchanged. Explain a possible cause.
3. Factories have spare capacity. Why need higher investment not cause a large immediate price rise? Name a leakage and distinguish it from an injection.
4. CPI rises from 125 to 130; weekly pay rises £600 to £612. Calculate inflation and the real-pay change.
5. Automation displaces packers while technician vacancies remain. Explain the unemployment type and evidence needed to compare the harm with a demand downturn.

Check after attempting. (1) Real-output factor $1.08/1.10 \approx 0.982$: real GDP falls about 1.82%. (2) Some jobseekers may leave the labour force. (3) Spare capacity allows more production with limited price pressure; later investment may expand supply. Saving, taxation and imports are leakages; investment, government spending and exports are injections. (4) Inflation is $[(130 - 125)/125] \times 100\% = 4\%$. Real-pay change is $[(612/600)/(130/125) - 1] \times 100\% \approx -1.92\%$. (5) Technological displacement can overlap with structural skill mismatch; compare numbers, duration, lost income and access to suitable jobs before ranking harm against cyclical unemployment.

Vocabulary card meanings

gross national income: Total income earned by a country's residents, at home and abroad. real: Adjusted so that rising prices do not inflate the figure. standard of living: The material comfort people can afford from their income. distribution: How income or output is shared among people. leakages: Income that leaves the circular flow through saving, tax, or imports. closed economy: An economy that does not trade with other countries. open economy: An economy that trades goods and capital with other countries. productive capacity: The most an economy can make with the resources it has now. real output: Production counted in constant prices, so inflation

is removed. slump: A deep, long fall in production and jobs. purchasing power: How much a given sum of money can buy. cost-push inflation: Rising prices caused by higher costs of making goods. value added: The extra worth a firm creates beyond the inputs it buys. net national income: Residents' income after wear and tear on capital is subtracted. capital depreciation: The loss of value in machines and buildings as they wear out. market prices: The prices buyers actually pay, including indirect tax. basic prices: The prices producers receive before indirect tax is added. technological unemployment: Job loss because a machine now does the work.

Government macroeconomic intervention

A-Level Economics

Governments use taxes, spending and monetary tools to influence activity. Begin with the problem: weak spending, rapidly rising prices and a skills mismatch need different explanations and responses.

Macroeconomic policy objectives

The core AS objectives in this unit are **price stability** , low unemployment and **economic growth** . Price stability means low, stable **inflation** ; use the target stated in a question rather than assuming every country has the same numerical target. Growth means rising real output. Check population and distribution before claiming higher output improves everyone's living standards.

Low unemployment means people seeking and available for work can find suitable jobs. **full employment** does not require zero measured unemployment: workers and vacancies take time to match. Read employment and participation alongside the unemployment rate.

Fiscal and monetary tools are often called **demand-side policies** because they influence **aggregate demand** . But policy labels can overlap: the design can also affect **aggregate supply** .

Worked explanation. Government purchases of port construction immediately raise G, a component of AD. If the finished port reduces transport delays, it can later raise productive capacity. Fiscal spending can therefore have a demand effect now and a supply effect later; the second effect depends on the project's quality and use.

Broader objectives include a sustainable current-account position within the **balance of payments** and **redistribution of income** . A possible **conflict** between objectives needs a causal explanation and conditions, not an automatic rule. These broader objectives and policy trade-offs are not required by AS syllabus section 5.1; the core work here is explaining the three policy types and their effects.

Fiscal policy

fiscal policy uses government taxation and spending. The **budget** records planned revenue and expenditure over a period. A **budget deficit** means expenditure exceeds revenue; a **budget surplus** means revenue exceeds expenditure. **national debt** is the outstanding stock of government borrowing, measured at a point in time. The annual deficit is a flow, not the debt itself.



Taxation and government spending are fiscal-policy tools.

■ Budget flows and the debt stock

Worked example. Illustrative annual spending is £120bn and revenue £100bn. Debt starts at £500bn. Assume the entire deficit is financed by net borrowing, with no other adjustments.

$$\text{deficit} = \text{spending} - \text{revenue}.$$

$$\text{deficit} = £120\text{bn/year} - £100\text{bn/year} = £20\text{bn/year}.$$

$$D_1 = D_0 + \text{net borrowing during year}.$$

$$D_1 = £500\text{bn} + £20\text{bn} = £520\text{bn}.$$

Next year spending is £118bn and revenue £110bn. Under the same financing assumptions:

$$\text{deficit} = \text{spending} - \text{revenue} = £118\text{bn} - £110\text{bn} = £8\text{bn}.$$

$$D_2 = D_1 + \text{net borrowing} = £520\text{bn} + £8\text{bn} = £528\text{bn}.$$

The deficit falls from £20bn to £8bn, but debt still rises. Actual debt also depends on financing choices and other adjustments. Debt servicing uses budget resources; whether borrowing is sustainable depends on the cost and capacity to service it, not just the debt's cash size.

Types and reasons for spending

capital spending buys or improves long-lived assets, such as new classroom computers or a new road. **current spending** supports ongoing activity, such as teachers' wages and rent for services this year.

A grant to a student is a transfer of income rather than a direct purchase of current output. Government purchases enter G; a transfer itself does not, although recipients may later spend it on consumption.

Spending can provide public services and infrastructure, improve access to education or support incomes. It may address market failure and support demand in a downturn. Judge benefits against resource costs and opportunity costs; not every public project automatically raises productivity.

Tax bases and income patterns

A **direct tax** is levied on income or profit, such as income tax or corporation tax. An **indirect tax** is levied on expenditure and collected through suppliers, such as a sales tax. These labels describe the tax base and collection route, not who ultimately bears the burden.

Taxes can raise revenue for services, influence demand, redistribute income or discourage activities with external costs. A tax on a harmful product may discourage consumption, but its effectiveness depends on demand response and alternatives.

A different classification asks how total tax as a share of income changes when income rises:

- **progressive tax** : the average rate rises.
- **proportional tax** : the average rate stays constant.
- **regressive tax** : the average rate falls.

Worked tax bands: total, average and marginal tax

The following annual tax system is hypothetical, not any country's current rules. There are no deductions. Only income within each band faces that band's rate.

Annual income band	Rate
First £10000	0%
Next £10000	10%
Above £20000	20%

At income £25000, the first £10000 faces 0%, the next £10000 faces 10%, and only the remaining £5000 faces 20%.

$$T = \sum_i t_i Y_i.$$

$$T = (0 \times £10000) + (0.10 \times £10000) + (0.20 \times £5000) = £2000.$$

The **average tax rate** (ATR) is the share of total income paid in tax:

$$\text{ATR} = \frac{T}{Y} \times 100\%.$$

$$\text{ATR} = \frac{£2000}{£25000} \times 100\% = 8\%.$$

At income £26000, total tax is £2200. The **marginal tax rate** (MTR) measures the tax on extra income. This £1000 increment stays inside the 20% band:

$$\text{MTR} = \frac{\Delta T}{\Delta Y} \times 100\%.$$

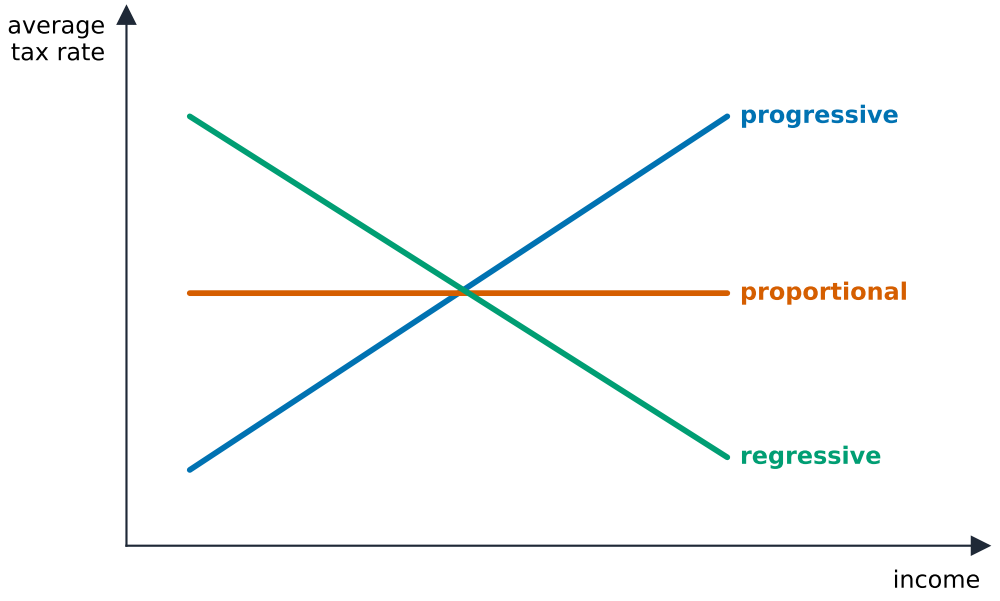
$$\text{MTR} = \frac{£2200 - £2000}{£26000 - £25000} \times 100\% = 20\%.$$

A marginal rate does not apply to all earlier income. If an income increment crosses bands, its extra tax must be calculated across the relevant bands.

Practice with solution. At income £15000, total tax is £500. Compare its ATR with the 8% rate at £25000:

$$\text{ATR} = \frac{T}{Y} \times 100\% = \frac{£500}{£15000} \times 100\% \approx 3.33\%.$$

The rising average rate makes this system progressive. A flat 10% income tax is proportional. A fixed £1000 charge per person is regressive: its income share falls from $1000/15000 \times 100\% \approx 6.67\%$ to $1000/25000 \times 100\% = 4\%$. Paying more pounds does not itself show a progressive tax. An indirect tax may be regressive relative to income when lower-income households spend a larger share on taxed goods; check their spending patterns and exemptions.

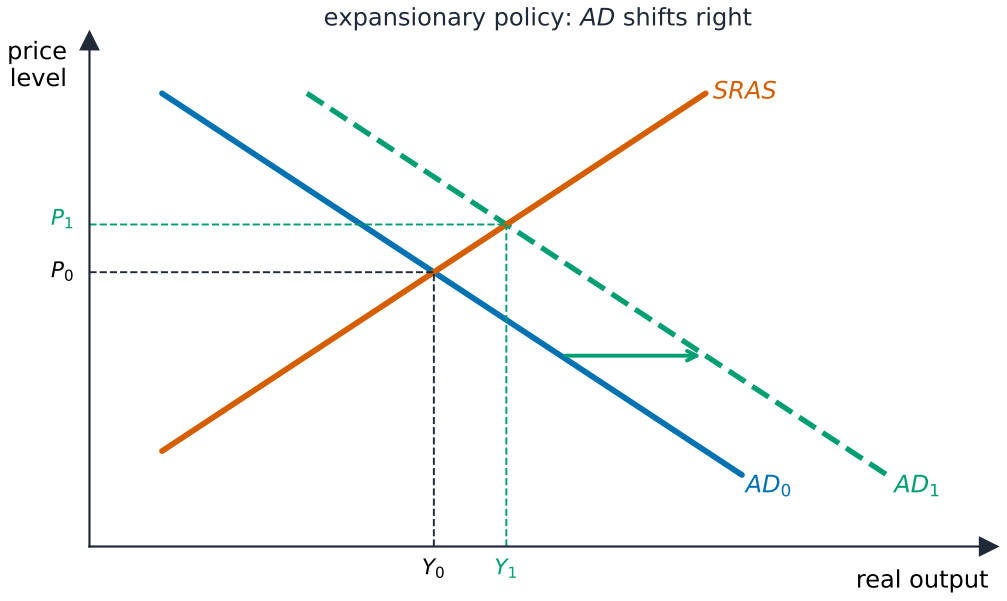


The classification uses the average tax rate, not the cash amount paid or the marginal rate alone.

■ Fiscal policy through AD and AS

expansionary fiscal policy aims to raise demand through higher government purchases or lower taxes. **contractionary** fiscal policy uses the reverse changes. A deficit alone does not prove a new discretionary expansion: weak activity can reduce revenue without a deliberate policy change.

Worked explanation. Compare an additional £10bn of government purchases of domestic services with a £10bn household income-tax cut. Purchases directly raise G in $AD = C + I + G + (X - M)$. The tax cut raises disposable income, but households may save part or buy imports, so extra domestic consumption need not equal £10bn. Identify the route to domestic planned spending instead of assuming equal effects.



With upward-sloping SRAS and other influences unchanged, an AD increase raises real output and the price level. Capacity conditions affect the balance of these changes.

With substantial spare capacity, extra demand can raise production with limited price pressure. Near capacity, demand mainly raises prices. Employment effects depend on how firms meet extra output, including changes in hours and productivity. Fiscal contraction reverses the demand pressure in this model, but it can reduce output and jobs as well as price pressure. Spending on useful infrastructure or education can have supply effects later.

■ Evaluate the constraint

Purchases can support demand directly, but planning and delivery may delay them. Income-tax cuts may support consumption, but cautious households may save the gain; imports can absorb part. The timing depends on the measure, not a rule that budgets can change only once a year.

Borrowing may add debt-service costs. **crowding out** can occur if government borrowing raises financing costs or public demand draws resources away from private activity. Its extent depends on credit conditions, spare resources and the policy response. It is not automatically large or absent. Match the judgement to the problem, financing and expected response rather than treating one policy as always best.

Monetary policy

monetary policy influences monetary and credit conditions through the **central bank** . The required tools are the **interest rate** , **money supply** and **credit regulation** (restrictions on lending).

Lower policy rates, an increased money supply in the simplified model, or easier lending restrictions can support spending. These are expansionary changes; contractionary changes work in the opposite direction. Banks must be willing to lend and customers willing to borrow for credit to support spending. In the simplified money-market model, an increase in money supply lowers interest rates with money demand unchanged; do not treat this as an unconditional prediction for every policy framework.



A central bank influences monetary and credit conditions; policy rates and commercial loan rates need not be identical.

■ Worked borrowing and credit examples

An illustrative £10000 variable-rate loan has no repayments or fees during the year. Its annual rate rises from 6% to 7%.

$$\text{annual interest} = \text{principal} \times \text{annual rate}.$$

$$\text{interest}_0 = £10000 \times 0.06 = £600/\text{year}.$$

$$\text{interest}_1 = £10000 \times 0.07 = £700/\text{year}.$$

The extra £100 annual cost may make some projects less attractive. An existing fixed-rate loan need not change immediately. Policy-rate changes influence commercial rates, but pass-through depends on contracts and banks.

A hypothetical borrowing limit falls from four times annual income to three times. With income £40000 per year, the maximum loan falls from £160000 to £120000. The income multiple represents a number of years of that annual income:

$$\text{maximum loan} = \text{income multiple} \times \text{annual income}.$$

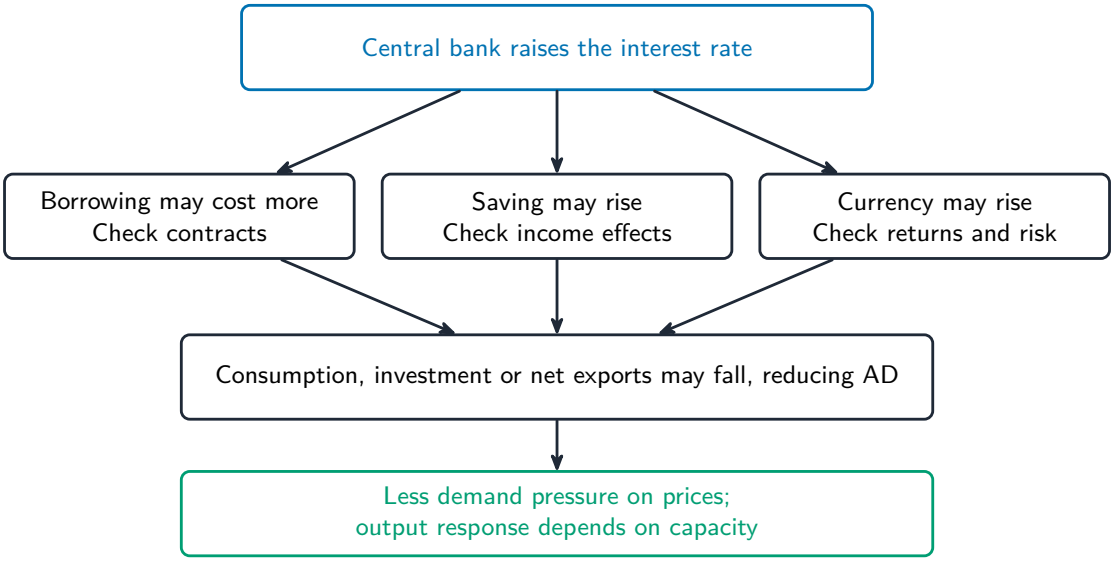
$$L_0 = 4 \text{ years} \times £40000/\text{year} = £160000.$$

$$L_1 = 3 \text{ years} \times £40000/\text{year} = £120000.$$

This tighter credit rule can limit spending even if the interest rate is unchanged. A borrower who did not need the extra credit may not change spending. Easing a limit does not force borrowing. These are teaching figures, not an actual country's lending rules.

The transmission mechanism

The **transmission mechanism** links policy to spending, then to output and prices. A higher interest rate can make borrowing dearer and saving more rewarding. Households may reduce credit-financed consumption and firms may reduce investment, but contracts, confidence and income effects affect the response.



These are possible responses, not guaranteed changes. Contracts, confidence, international conditions and spare capacity affect the strength and timing.

If domestic rates rise relative to foreign rates, higher expected returns may attract funds and increase demand for the currency. A stronger **exchange rate** can make exports dearer to foreign buyers and imports cheaper domestically, weakening

net exports and AD. But expectations, risk and changes abroad also affect capital flows. Trade quantities depend on contracts and responsiveness to prices. An interest-rate rise alone does not prove that exports must fall.

Effective monetary expansion may raise C or I and shift AD right, like the fiscal AD increase illustrated earlier. With upward-sloping SRAS, real output and price level rise, other things unchanged. Near capacity, price pressure dominates; with spare capacity, output can respond more. Employment depends on how firms meet extra output, including hours and productivity. Rate decisions can change quickly, but borrowing, spending and production take time; there is no fixed lag for every economy or instrument.

Worked comparison. Factories have idle capacity, confidence is weak and policy rates are near zero. Extra public purchases can support demand directly, but delivery, funding and imports affect their impact. Weak confidence may limit borrowing even when credit is cheap; near-zero rates restrict further conventional rate cuts. This does not prove every monetary tool is exhausted or that fiscal action must succeed. Compare feasible measures, their likely response and timing. A different starting interest rate alone does not determine the best policy.

Supply-side policy

supply-side policy aims to raise **productivity** and **productive capacity**, shifting LRAS right. Productivity is useful output per unit of input; raising it need not mean employing more people.

■ Worked productivity example

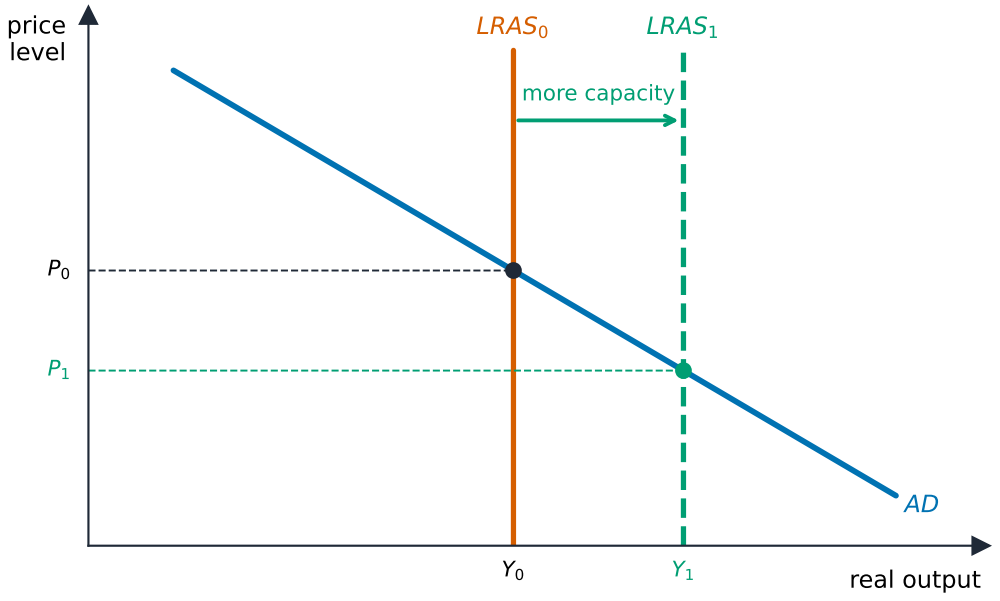
In an illustrative factory, 100 workers each work 8 hours per day. Useful training and new machinery raise output per worker-hour from 10 to 12 units, with workers and hours unchanged.

$$\text{daily output} = \text{workers} \times \text{hours per worker-day} \times \text{units per worker-hour}.$$

$$Q_0 = 100 \times 8 \times 10 = 8000 \text{ units/day}.$$

$$Q_1 = 100 \times 8 \times 12 = 9600 \text{ units/day}.$$

Possible output rises by 20%, with no extra workers in this case. Actual output and sales also require demand, inputs and use of the improved process.



In this model, higher capacity supports higher real output at a lower price level. A comparison of price levels does not prove a continuing fall in inflation every year.

■ Direct provision: skills, infrastructure and technology

training can improve relevant skills and reduce a mismatch between jobseekers and vacancies. Suppose an industry closes while local technician vacancies remain. Of 140 trainees, 120 qualify and 90 find suitable jobs. Qualifications alone do not create vacancies: location, demand and access matter. Check completion, matching and whether jobs are additional rather than displaced from other applicants. Retraining alone need not restore sales during a general demand slump.

Better **infrastructure** can reduce transport delays, improve access to suppliers and lower unit costs. Check reliability, use and the value of saved resources against construction costs. Research support can improve production methods, but success and adoption are uncertain. Funding alone does not prove higher productivity. Construction spending can raise AD before supply benefits arrive; time and quality vary by measure.

■ Market-based approaches

Lower income tax can increase the reward for extra work or investment, changing the **incentive** . The response depends on available jobs, finance and people's preferences; some may value more leisure. Lower benefits may change job-search incentives but do not create suitable vacancies by themselves. Consider access and hardship as well as incentives.

privatisation transfers state enterprises to private ownership. It may strengthen incentives to control costs, but ownership alone does not create competition or better management. **deregulation** can remove unnecessary entry

barriers and encourage competition. Removing useful safety or environmental rules can also impose costs.

■ Comparing tools and evaluating claims

Worked comparison. Region A has workers with unsuitable skills and poor transport. Region B has qualified workers and efficient transport but few orders. Direct training and infrastructure may address A's gaps if useful and used. Tax incentives and easier entry can encourage work and investment, but do not automatically supply missing skills or transport. B's immediate constraint is weak demand, so extra capacity may remain unused. Compare cause, response, cost and timing; combinations may complement each other.

A useful supply improvement can support higher output with less price pressure in the stated model. It is not a promise of no costs. Funding uses resources, taxes can change spending and incentives, and gains may arrive unevenly or late. Judge quality, opportunity costs, distribution, demand and timing. Distinguish a lower price level in a diagram from a sustained change in the inflation rate.

Policy judgement and retrieval

Diagnose rising prices. Economy A has strong spending near capacity. B has a fuel-cost shock and weak sales. Monetary tightening may restrain demand pressure in A. In B, weaker demand does not directly remove the fuel shortage and may reduce output further. Supply measures may reduce costs but may take time or fail to address the specific shock. Compare source, persistence and feasible response rather than choosing an instrument from inflation alone.

Improve the assessment. "Training reduces unemployment, but there is a time lag" is incomplete. Explain that training can reduce a skills mismatch if vacancies need the new skills. Completion and matching take time, so lost income may persist. If firms lack orders, demand support may address the immediate sales constraint more directly. A judgement compares mechanisms, scale, time and costs; merely naming a lag is not complete evaluation.

Attempt before checking. Spending is £90bn/year and revenue £84bn/year; starting debt is £300bn. Assume the deficit is financed by net borrowing with no other adjustments. Find deficit and closing debt. A worker pays £3000 tax on £30000 income, then £3200 on £31000: find ATR and MTR on the extra income.

$$\text{deficit} = \text{spending} - \text{revenue} = £90\text{bn/year} - £84\text{bn/year} = £6\text{bn/year}.$$

$$D_1 = D_0 + \text{net borrowing} = £300\text{bn} + £6\text{bn} = £306\text{bn}.$$

$$\text{ATR} = \frac{T}{Y} \times 100\% = \frac{£3000}{£30000} \times 100\% = 10\%.$$

$$\text{MTR} = \frac{\Delta T}{\Delta Y} \times 100\% = \frac{£3200 - £3000}{£31000 - £30000} \times 100\% = 20\%.$$

Final check. Explain a monetary tool other than interest rates. Explain why training can raise capacity without guaranteeing jobs. Why can a government package that raises income tax and funds apprenticeships be both fiscal and supply-side?

Check after attempting. Easing credit regulation may allow more financed spending if banks and borrowers respond; increased money supply may lower rates in the simplified model with unchanged money demand. Training can improve productivity or matching, but suitable vacancies and demand remain necessary. Tax and purchases are fiscal tools, while useful apprenticeships can also improve skills and capacity. Explain demand and supply effects separately and state their time periods.

Vocabulary card meanings

redistribution of income: Shifting income from some households to others, often by tax. demand-side policies: Government actions aimed at the level of total spending. privatisation: Selling firms owned by the state to private owners. deregulation: Removing rules that limit how firms may compete. capital spending: Public money spent on long-lived assets such as roads. current spending: Public money used up within the year, such as wages. average tax rate: Tax paid divided by total income. marginal tax rate: The share taken in tax from one extra unit of income. credit regulation: Rules that limit how much households and firms may borrow.

International economic issues

A-Level Economics

Why countries trade

international trade lets residents exchange goods and services across borders. To **specialise** means to concentrate on producing particular goods or services. Specialisation and trade can increase consumption choices; calculate the possible gain and then consider costs and adjustment.



A container port: trade connects producers and buyers across borders.

Absolute and comparative advantage

An **absolute advantage** means producing more with the same stated resources. A **comparative advantage** means producing at lower **opportunity cost**. They answer different questions.

Illustrative maximum outputs with the same quantity of resources in each country; assume straight-line PPCs and unchanged technology:

Country	Wine only	Cloth only
A	20 units	10 units
B	5 units	5 units

A has an absolute advantage in both. To find comparative advantage, divide output forgone by output gained:

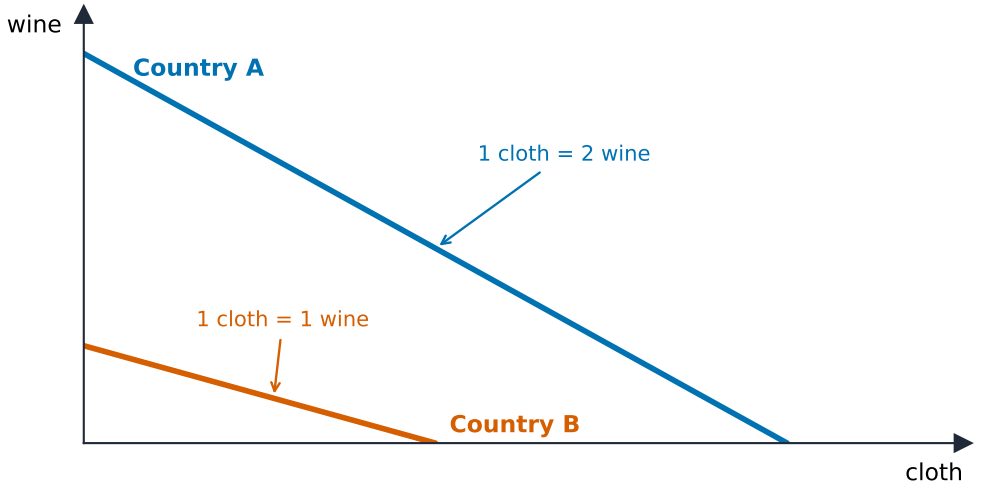
$$\text{OC}_A(\text{cloth}) = \frac{\text{wine forgone}}{\text{cloth gained}} = \frac{20}{10} = 2 \text{ wine/cloth}.$$

$$\text{OC}_B(\text{cloth}) = \frac{\text{wine forgone}}{\text{cloth gained}} = \frac{5}{5} = 1 \text{ wine/cloth}.$$

$$\text{OC}_A(\text{wine}) = \frac{\text{cloth forgone}}{\text{wine gained}} = \frac{10}{20} = 0.5 \text{ cloth/wine}.$$

$$\text{OC}_B(\text{wine}) = \frac{\text{cloth forgone}}{\text{wine gained}} = \frac{5}{5} = 1 \text{ cloth/wine}.$$

B gives up less wine per cloth, so has comparative advantage in cloth. A gives up less cloth per wine, so has comparative advantage in wine. One country being more productive in both goods does not remove possible gains from trade.



comparative advantage = lower opportunity cost

Different slopes show different opportunity costs: A gives up 2 wine per cloth; B gives up 1.

Gains in consumption and the trading possibility curve

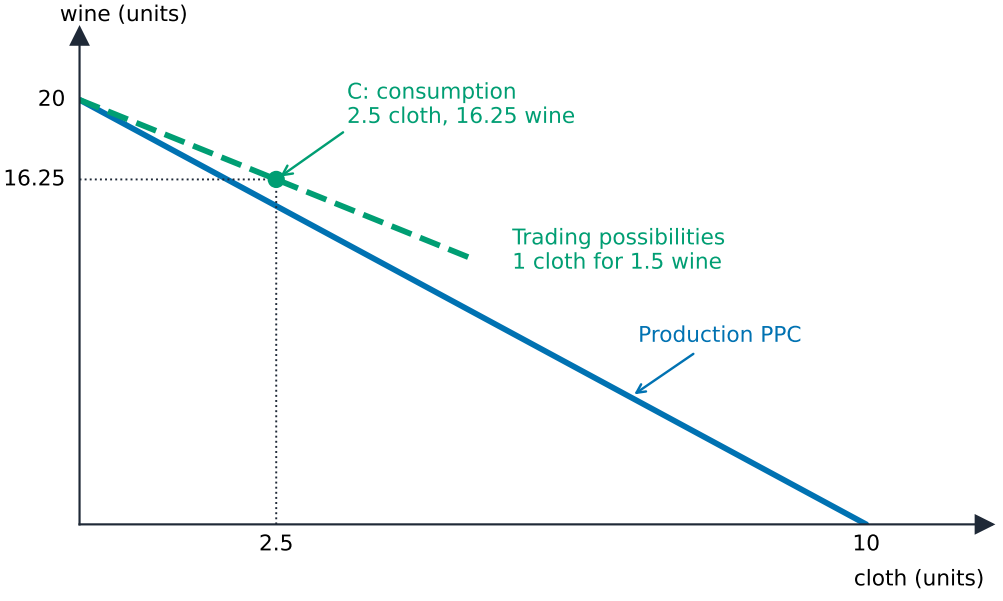
Worked exchange ratio. B gives up 1 wine to make 1 cloth; A gives up 2 wine to make it. At 1 cloth for 1.5 wine, B receives 0.5 wine more than its cost and A saves 0.5 wine compared with making cloth at home. Both strictly gain per traded unit when the wine-per-cloth ratio is between 1 and 2, before trading costs. At an endpoint, one side is indifferent in this model.

Worked consumption bundles. Without trade, A makes 16 wine and 2 cloth; B makes 3 wine and 2 cloth. Both lie on their own PPCs. After specialisation, A makes 20 wine and B makes 5 cloth. B sends 2.5 cloth to A in exchange for 3.75 wine, consistent with 1.5 wine per cloth.

Country	Wine consumed	Cloth consumed
A	$20 - 3.75 = 16.25$	$0 + 2.5 = 2.5$
B	$0 + 3.75 = 3.75$	$5 - 2.5 = 2.5$

Both can consume more of both goods than in the stated no-trade bundles. Total consumption equals total production: 20 wine and 5 cloth. The example shows a possible gain, not a promise that all trades or all households gain equally.

A **trading possibility curve** shows consumption combinations available through production and trade at a stated exchange ratio. It can extend beyond a country's production PPC without moving that PPC. The country's technology and resources have not changed; trade changes access to goods. Here B can supply at most 5 cloth, so the shown trading segment is limited to those imports.



At 2.5 cloth, A's own production PPC permits 15 wine; the agreed trade permits consumption of 16.25 wine. This is a consumption gain, not a shift in production capacity.

■ Terms of trade: prices, not the trade balance

terms of trade compare export prices with import prices, using indices with the same base period:

$$\text{ToT} = \frac{P_X}{P_M} \times 100.$$

With export-price index 110 and import-price index 105:

$$\text{ToT} = \frac{110}{105} \times 100 \approx 104.8.$$

A fixed export quantity now buys about 4.8% more imports than in the base period. The index does not measure export volumes or the current-account balance. An improvement could reflect higher export prices or lower import prices, with different effects on sales and costs. Commodity demand, supply and production costs influence the ratio. Dearer exports may sell fewer units; improved terms of trade do not automatically mean higher export revenue or welfare.

Transfer task with solution. Terms of trade fall from 95 to 85 while the import-price index stays 120. Find both export-price indices and their change:

$$\text{ToT} = \frac{P_X}{P_M} \times 100 \quad \Rightarrow \quad P_X = \frac{\text{ToT} \times P_M}{100}.$$

$$P_{X0} = \frac{95 \times 120}{100} = 114.$$

$$P_{X1} = \frac{85 \times 120}{100} = 102.$$

$$\Delta P_X = P_{X1} - P_{X0} = 102 - 114 = -12.$$

Export prices fall 12 index points, not 10. Distinguish an index-point change from a percentage change.

■ Limits and distribution of gains

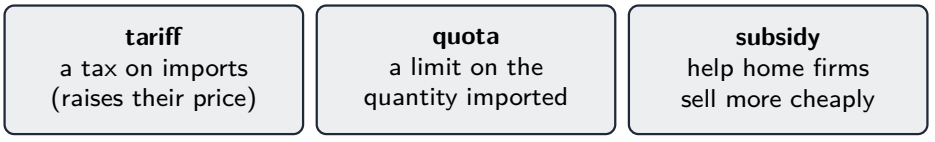
If the calculated saving is 0.5 wine per cloth but shipping and border costs use resources worth 0.6 wine per cloth, the saving is lost. The simple model holds technology and resources fixed, assumes constant opportunity costs, and omits trading costs and barriers. Moving workers and capital into expanding industries can require time, retraining and investment. Changing costs, product quality and reliance on one export also matter. Trade can widen choice and competition; assess adjustment losses and the distribution of gains instead of assuming everyone benefits equally.

Protectionism

protectionism shields home producers from foreign competition. Explain each tool's direct mechanism before judging its effects:

- A **tariff** taxes imports. In the small-country model below, it raises the domestic price while imports remain available.
- A **quota** caps the quantity imported. A quota above existing imports may not bind.
- An export **subsidy** pays exporters, reducing effective costs and potentially expanding foreign sales. It requires public funding; it does not automatically raise every domestic price.

- An **embargo** bans specified trade, potentially increasing the cost of alternatives.
- **administrative barriers** are excessive procedures that hinder imports. Distinguish these from useful safety or environmental standards.



Tariffs change import costs, quotas limit import quantities, and subsidies support specified producers. Their revenue and spending effects differ.

■ Worked tariff: imports and revenue

Assume a small country facing an unchanged world price, no evasion and full pass-through of the tariff into the domestic price. Prices are £/unit; quantities are million units/year. At world price 4, domestic supply is 2 and demand 8. A tariff of 2 raises the domestic price to 6: supply rises to 4 and demand falls to 6.

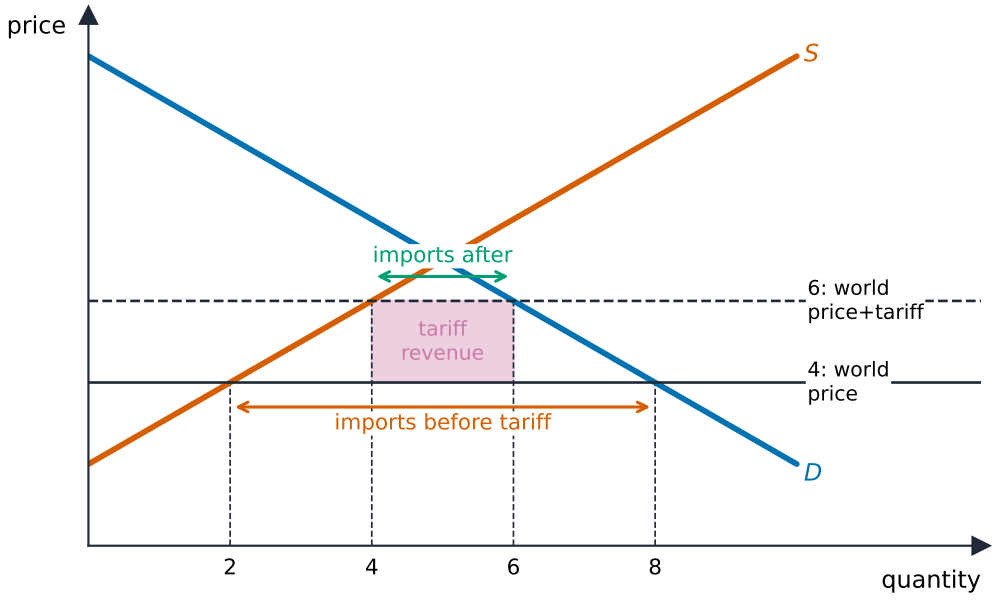
$$M_0 = Q_{d0} - Q_{s0} = 8 - 2 = 6 \text{ million units/year.}$$

$$M_1 = Q_{d1} - Q_{s1} = 6 - 4 = 2 \text{ million units/year.}$$

$$R = t \times M_1.$$

$$R = £2/\text{unit} \times 2 \text{ million units/year} = £4 \text{ million/year.}$$

Use imports after the tariff. Multiplying the tariff by all domestic consumption or the old import quantity gives the wrong revenue.



Under these assumptions, imports fall from 6 to 2 million units/year; the rectangle is the tariff per unit multiplied by remaining imports.

Effects on stakeholders

A **stakeholder** is a person or group affected. Consumers pay 6 rather than 4 and buy 6 rather than 8 million units, reducing purchasing power. Home producers sell 4 rather than 2 million units at the higher price, but higher-cost production may expand. Government receives £4 million/year while taxable imports continue; revenue is a transfer, not evidence of a net national gain. Foreign suppliers sell fewer units in this model. Domestic firms using imported inputs can face higher costs. Compare the gains and losses and wider responses.

A quota need not give gains to the same recipient

Suppose an import quota of 2 million units/year produces the same domestic price 6 while foreign supply remains available at 4. A licence holder buying at 4 and selling at 6 may receive the difference. Government can capture some or all of it by auctioning licences; foreign suppliers may receive gains if they hold licences or market power. Do not claim that every quota generates no government receipts or always benefits domestic importers. A tariff fixes a tax per imported unit; a quota fixes the permitted quantity, so their response to changing demand can differ.

Evaluate arguments for and against protection

An **infant industry** may need time to learn or reach efficient scale. A credible temporary-protection case needs evidence of falling costs, measurable targets and an end date. Protection may instead preserve inefficiency and impose costs on consumers. Strategic capacity and time for workers to adjust may also matter.

dumping concerns export prices below normal value, commonly compared with home-market prices or assessed using costs. Cheap imports alone do not prove an intention to destroy competitors. WTO definition.

retaliation by trading partners can weaken export sales. Judge a specified tool against its objectives, duration, consumer and input costs, possible foreign responses and alternatives. A justified conclusion weighs the actual conditions; it does not simply repeat that protection saves jobs or raises prices.

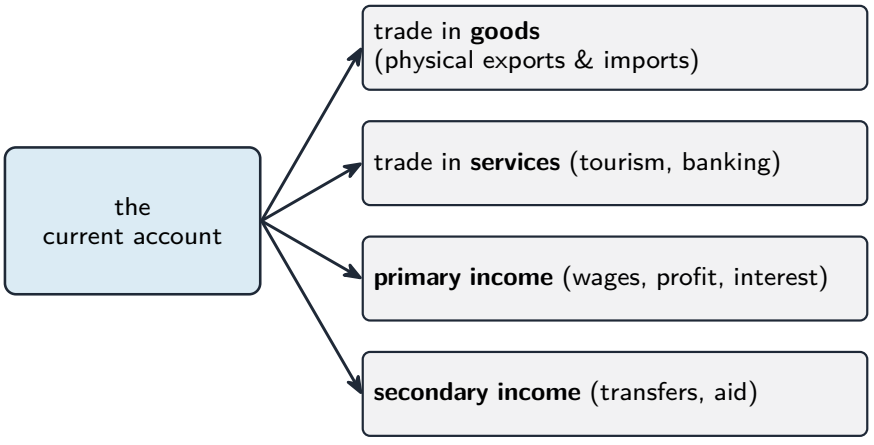
The balance of payments

The **balance of payments** records transactions between residents and non-residents during a period. The **current account** is one part; it records goods, services, primary income and secondary income. A receipt is a credit, and a payment is a debit. Use the same period, currency and sign convention throughout a calculation.

■ Classify transactions before calculating

- **trade in goods** : exports and imports of physical products. A resident firm exporting machinery receives a goods credit.
- **trade in services** : services supplied across the resident/non-resident boundary. A foreign visitor paying a resident hotel creates a service credit.
- **primary income** : income from labour or ownership of assets across borders. Interest received by a resident on a foreign bond is a credit; interest paid abroad is a debit.
- **secondary income** : current transfers without an equivalent item received in return, such as a household gift from abroad.

Buying an existing foreign bond is a financial-asset transaction, not primary income. Its later interest is income. Goods made and consumed domestically are not exports simply because they are valuable.



Calculate the net balance of each component; then add all four, including income and transfers.

■ Worked account: trade surplus but current-account deficit

Illustrative annual flows, all £bn/year:

Component	Receipts	Payments
Goods	80	100
Services	50	20
Primary income	5	20
Secondary income	2	5

For goods and services, receipts are export earnings and payments are import expenditure. Let B_g and B_s be their balances, B_t the combined trade balance, and N_p and N_s net primary and secondary income. For income, R means receipts and P payments.

$$B_g = X_g - M_g = 80 - 100 = -20.$$

$$B_s = X_s - M_s = 50 - 20 = +30.$$

$$B_t = B_g + B_s = -20 + 30 = +10.$$

Goods are in deficit, services in surplus, and combined trade in surplus. Income and transfers still matter:

$$\begin{aligned} N_p &= R_p - P_p = 5 - 20 = -15 \\ N_s &= R_s - P_s = 2 - 5 = -3 \\ \text{CAB} &= B_t + N_p + N_s \\ \text{CAB} &= 10 + (-15) + (-3) = -8 \text{ (£bn/year)}. \end{aligned}$$

A **current account deficit** means total current-account debits exceed credits. A **current account surplus** means credits exceed debits. Neither is defined solely by imports and exports. This example has a £10bn trade surplus alongside an £8bn current-account deficit.

■ Explain the cause before choosing a policy

Transfer task with solution. Starting from CAB $= -8$, compare an extra £12bn/year of service exports with an extra £12bn/year of interest payments abroad. Hold all other flows constant.

$$\text{CAB}_{\text{exports}} = \text{CAB}_0 + \Delta X_s = -8 + 12 = +4 \text{ (£bn/year)}.$$

$$\text{CAB}_{\text{interest}} = \text{CAB}_0 - \Delta P_p = -8 - 12 = -20 \text{ (£bn/year)}.$$

Stronger overseas demand can raise exports; stronger domestic spending can raise imports. Exchange rates, relative prices, product quality and capacity influence trade. Income payments and transfers can change CAB even when trade is unchanged. A single balance does not identify its cause.

■ Assess consequences and financing

Compare two economies with the same deficit. Economy A imports machinery for a viable new export industry. Economy B persistently loses overseas customers while paying more income abroad. A's imports may improve future capacity and earnings if the investment succeeds. B's lost sales may weaken demand and employment, although other expenditure can offset this. Size alone cannot settle the judgement.

A deficit requires counterpart financing or asset/reserve changes in the wider accounts. Borrowing, equity investment and selling assets differ: external debt does not automatically rise by the current-account deficit. Consider persistence, financing terms, future income payments, exchange-rate pressures and productive use of imports. A surplus can reflect strong competitiveness or weak domestic spending; it is not automatically a welfare gain.

Exchange rates

An **exchange rate** is the price of one currency in another. State the quote before describing a rise or fall. For example, US1.40*perpoundmeans*£1*buysUS*1.40. If this rises to US$1.60 per pound, the pound buys more dollars: **appreciation** . A fall in the dollar-per-pound quote is **depreciation** . A quote in pounds per dollar moves the opposite way.



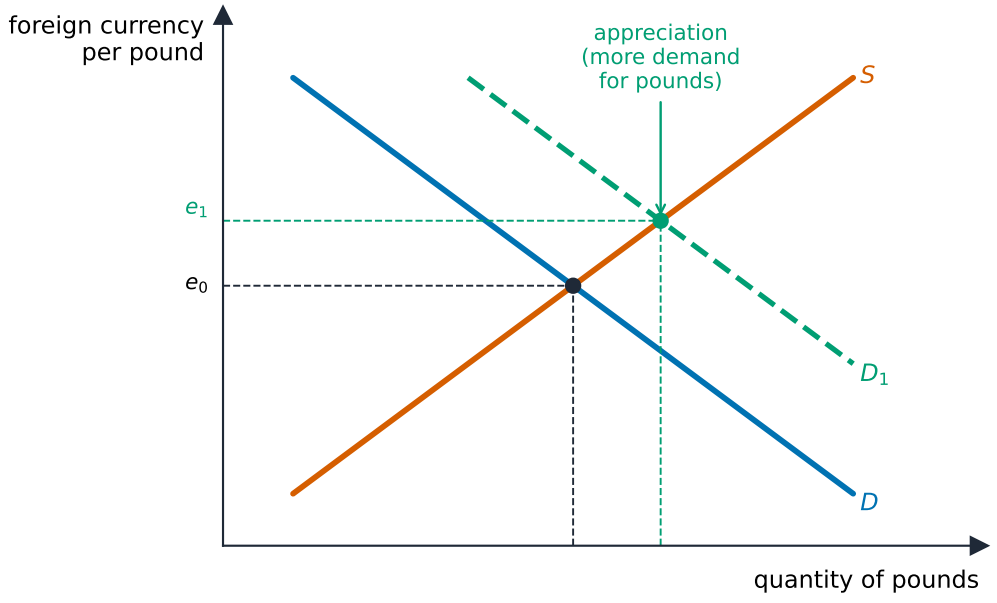
Read the currencies and quote direction before interpreting a number.

The floating currency market

A **floating exchange rate** adjusts where demand for the currency meets supply. In the market for pounds, a buyer exchanging dollars for pounds to pay a UK exporter demands pounds. A UK importer exchanging pounds for dollars supplies pounds. Follow the currency exchanged, not the buyer's nationality.

With currency supply unchanged, stronger demand for pounds raises foreign currency per pound: appreciation. Higher export demand or inward investment may raise demand. With currency demand unchanged, UK residents buying more foreign goods can increase supply of pounds and cause depreciation. Outward investment can also increase supply. Higher relative interest returns may attract funds, but expectations and risk can offset that response.

A change in the pound's own price causes movement along a curve; a change in export demand or investment flows can shift a curve. State other influences held constant.



The new equilibrium lies on both the shifted demand curve and unchanged supply curve. This is a conditional demand-shift example, not a forecast of every exchange-rate change.

For context, a **fixed exchange rate** is maintained at a stated rate or band using intervention or other policy measures. Maintaining it can require reserves and policy adjustments. A **managed exchange rate** combines market forces with intervention to influence the rate. These arrangements describe currency pricing; a current-account surplus describes a different account and does not identify the currency system. The required AS model here focuses on floating rates.

■ Worked export conversion

A UK machine keeps its £200 price. The rate rises from US1.40/£*toUS*1.60/£. Assume the change is passed through to the US buyer. Let P_{US} be its dollar price, P_{UK} its pound price and e dollars per pound. Multiply so pounds cancel:

$$P_{\text{US}} = P_{\text{UK}} \times e.$$

$$P_{\text{US},0} = P_{\text{UK}} \times e_0 = £200 \times \text{US\$1.40/£} = \text{US\$280}.$$

$$P_{\text{US},1} = P_{\text{UK}} \times e_1 = £200 \times \text{US\$1.60/£} = \text{US\$320}.$$

The machine becomes dearer in dollars. Sales may fall, depending on demand, contracts, quality and how firms adjust prices. Conversion arithmetic alone does not establish the quantity response.

Worked import conversion

A UK firm owes a fixed US$700 invoice. Use the same rates. Dividing dollars by dollars per pound gives pounds:

$$P_{\text{UK}} = \frac{P_{\text{US}}}{e}.$$

$$P_{\text{UK},0} = \frac{P_{\text{US}}}{e_0} = \frac{\text{US\$700}}{\text{US\$1.40/£}} = £500.$$

$$P_{\text{UK},1} = \frac{P_{\text{US}}}{e_1} = \frac{\text{US\$700}}{\text{US\$1.60/£}} = £437.50.$$

Appreciation lowers this invoice's pound cost. Always check the units rather than memorising that imports mean multiplication or division.

Price effects do not settle the trade balance

Transfer task with solution. The pound depreciates from US1.40/£*toUS*1.20/£. A fixed £200 export now costs US240, *whileafixedUS*700 import costs about £583.33:

$$P_{\text{US}} = P_{\text{UK}} \times e = £200 \times \text{US\$1.20/£} = \text{US\$240}.$$

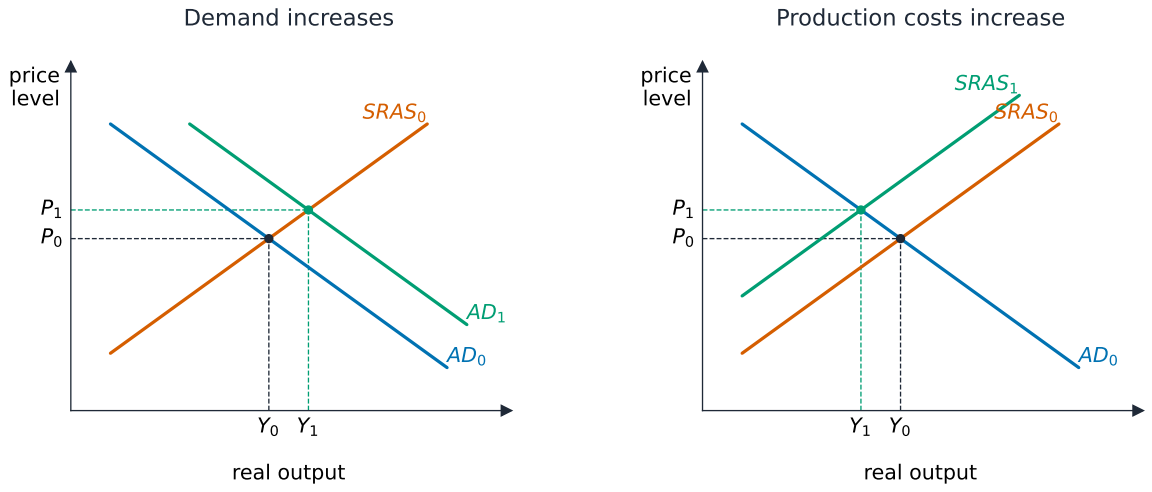
$$P_{\text{UK}} = \frac{P_{\text{US}}}{e} = \frac{\text{US\$700}}{\text{US\$1.20/£}} \approx £583.33.$$

Export receipts and import spending equal price times quantity. Buyers may be tied to contracts, essential imports may be hard to replace, and exporters may lack capacity. Cheaper exports do not automatically mean higher receipts; dearer imports do not automatically mean a lower import bill. Assess demand responsiveness, timing and the currency in which values are measured. The whole current account also includes income and transfers. A short technical label alone cannot replace this reasoning.

AD/AS: demand and imported costs

A depreciation can raise net exports if quantities respond, increasing **aggregate demand** (AD). With upward-sloping short-run supply and other influences unchanged, AD shifting right raises real output and the price level. Employment also depends on how firms meet extra demand, including hours and productivity.

Dearer imported fuel, components or machinery can increase production costs. Higher costs shift **short-run aggregate supply** (SRAS) left: holding AD unchanged, real output falls and the price level rises. If demand and cost effects occur together, their relative size determines the output change. Depreciation does not automatically remove structural unemployment or guarantee higher output and jobs.



Two conditional channels: each panel holds the other curve unchanged. Do not add their output effects as if both pointed the same way.

Correcting a current account deficit

First identify the component causing the imbalance, its persistence and financing. The aim of stability does not mean forcing a zero balance every year at any cost. Compare outcomes for growth, jobs, inflation and future income payments.

■ Fiscal and monetary policies: reducing expenditure

expenditure-reducing policies lower domestic demand. Higher income taxes or lower government purchases may reduce spending and imports. If demand is already weak, output and employment can also fall. The import response depends on which spending changes and the availability of domestic alternatives.

Higher interest rates may reduce borrowing and spending, including imports. They may also attract capital and appreciate the currency, making exports dearer abroad and imports cheaper at home. Expectations and risk matter. Explain both channels before claiming that tighter monetary policy must improve CAB.

■ Protection and relative prices: switching expenditure

expenditure-switching changes relative prices to favour domestic goods. A tariff may reduce imports; an export subsidy may support foreign sales but directly costs public funds. A tariff's receipts and a subsidy's costs have different effects on the government budget. A quota changes permitted quantities and its revenue effects depend on licence arrangements.

Depreciation can also change relative prices, but a floating rate is not simply set lower by government command. Monetary policy can influence it with uncertain responses. Higher import costs, retaliation, weak demand responses and limited capacity can offset gains. Compare a specified policy with feasible alternatives rather than promising

all effects from its label.

■ Supply-side policies: address the actual constraint

Supply-side policy can improve **competitiveness** through productivity, quality, costs and reliability. Compare two exporters: A cannot deliver reliably because a port is congested; B has spare capacity but foreign buyers no longer want its product. Port investment may address A's constraint; B may need product development or a different market. The same training programme need not solve both.

Skills, infrastructure and technology use resources and may take time. Higher capacity does not guarantee sales. Judge opportunity cost, the actual bottleneck and overseas demand before predicting higher exports.

■ Diagnose before giving a conclusion

Worked judgement. CAB is −£8bn/year despite a £10bn trade surplus because net primary and secondary income together are −£18bn/year. A tariff targets imports, not the income payments causing the imbalance. It could still change trade, but consumer and input costs, the size of the response and retaliation matter. Recommend a policy using the identified cause and compare benefits with other objectives.



Trade is only part of the current account: include net income and transfers when diagnosing an imbalance.

Apply and review

Attempt before checking. Annual balances, in £bn/year: goods−12, services+8, primary income−7, secondary income+1. Find CAB. Then £1 falls from US1.50*toUS*1.20. Find the dollar price of a fixed £150 export. Does the lower foreign price prove that CAB will improve?

Solution. Add all four components:

$$\text{CAB} = B_g + B_s + N_p + N_s = -12 + 8 - 7 + 1 = -£10\text{bn/year}.$$

$$P_{\text{US}} = P_{\text{UK}} \times e = £150 \times \text{US\$1.20/£} = \text{US\$180}.$$

Previously the export cost US$225. The pound depreciates; export quantities, import costs, income and transfers still determine the overall account.

Improve a judgement. Replace "depreciation will solve the deficit and unemployment, so it is the best policy" with a conditional explanation. A lower foreign price can support export demand if buyers respond and firms can expand. Dearer inputs may increase costs and prices, while essential imports can keep the import bill high. Judge the cause of the imbalance, demand responsiveness, capacity and timing, then compare alternatives that address the problem. A justified conclusion follows those conditions instead of promising that one tool meets every objective.

Vocabulary card meanings

international trade: Buying and selling across national borders. embargo: A government ban on trade with a named country or product. administrative barriers: Paperwork and rules that make imports harder without using a tariff. infant industry: A new home industry not yet able to match foreign rivals. current account deficit: Paying more to other countries for trade and income than is received. current account surplus: Receiving more from other countries for trade and income than is paid. managed exchange rate: A currency price that the central bank steers rather than leaving fully free. trading possibility curve: The combinations of goods a country can enjoy once it specialises and trades.

Business and its environment

A-Level Business

Why businesses exist

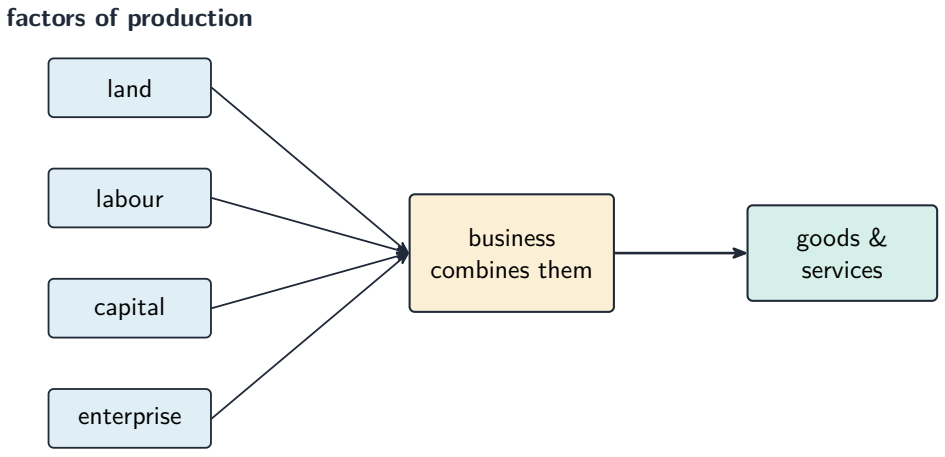
People have **needs** (things you must have, like food and shelter) and **wants** (things you would like, like a phone or a holiday). A business exists to meet these needs and wants by making **goods** and **services**.



Shops competing on a high street – the market a business operates in.

To do this, a business uses **resources**. We sort resources into four **factors of production**:

- **land** – natural things, such as soil, water and minerals.
- **labour** – the work done by people.
- **capital** – the machines, tools, buildings and money used to produce.
- **enterprise** – the skill of bringing the other three together and taking the **risk** to start a business.



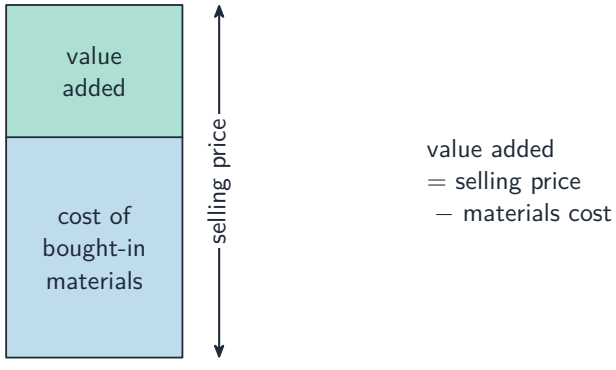
Land, labour, capital and enterprise are the four factors a business combines to make goods and services

Resources are **scarce** – there are not enough to meet everyone's wants. So every business must choose how to use them well.

Adding value

Adding value means the selling price of a product is higher than the cost of the **inputs** (the bought-in materials) used to make it.

$$\text{value added} = \text{selling price} - \text{cost of bought-in materials}$$



Value added is the selling price minus the cost of bought-in materials

A business can add value by **branding**, better quality, good design, fast service, or convenience. More value added can mean more **profit**, but only if costs stay under control.

Entrepreneurs and intrapreneurs

An **entrepreneur** is a person who takes the risk to set up and run a business. They have the idea, organise the factors of production, and make the decisions. If the business fails, they can lose their money.

An **intrapreneur** is an employee inside a larger business who acts like an entrepreneur. They build new ideas, products or ways of working, using the company's resources.



A food truck is enterprise you can see. The owner spotted a gap, spent their own savings on the van and stock, and now takes the daily risk of running it – the essence of being an entrepreneur

■ Characteristics of successful entrepreneurs

Successful entrepreneurs are usually:

- **innovative** – full of new ideas.
- willing to take risks, but in a careful way.
- hard-working and **determined** – they do not give up easily.
- **self-confident** and good at making decisions.
- good leaders who can plan ahead and manage money.

■ Why enterprise matters to a country

When people start new businesses (**business start-ups**), the whole country can gain:

- new jobs are created, which lowers **unemployment** .
- more goods and services are made, which raises **output** (and the country's **GDP**).

- more **competition** , giving customers lower prices and more choice.
- some firms grow and **export** , bringing money into the country.
- **economic growth** and a higher standard of living.

The business plan

A **business plan** is a written document that describes the business idea and how it will work. It usually contains:

Part of the plan	What it covers
The idea	the product, and what makes it different
The market	target customers, market size, competitors
Marketing	price, promotion and place
Operations	how the product is made or delivered
Finance	start-up costs, a **cash-flow forecast** , expected profit
People	the owners and key staff

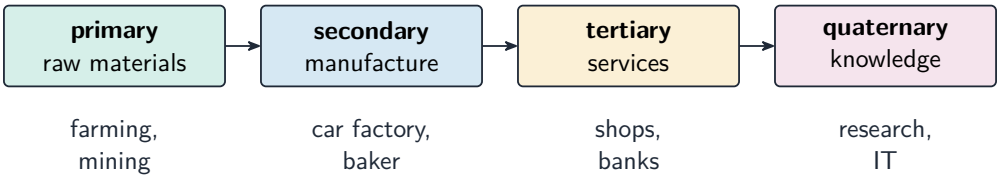
A business plan is useful because it helps the owner:

- get **finance** from a bank or an **investor** .
- set clear goals and lower the risk of failure.
- guide decisions as the business grows.

Sectors of business activity

Every business works in one of four sectors:

Sector	What it does	Example
primary sector	takes raw materials from nature	farming, mining, fishing
secondary sector	makes (manufactures) goods	car factory, baker
tertiary sector	provides services	shops, banks, transport
quaternary sector	knowledge and information services	research, IT, consulting



Business activity runs from the primary sector through to the quaternary sector

Here **raw materials** are the natural inputs a business starts with, and to **manufacture** means to make goods in large amounts. As a country develops, it usually moves from mostly primary work towards more tertiary and quaternary work.

Private and public sectors

- The **private sector** is made up of businesses owned by private people. Their main aim is usually profit.
- The **public sector** is made up of organisations owned by the government, such as state schools and public hospitals. Their aim is usually to provide a service, not to make profit.

Legal structures of business

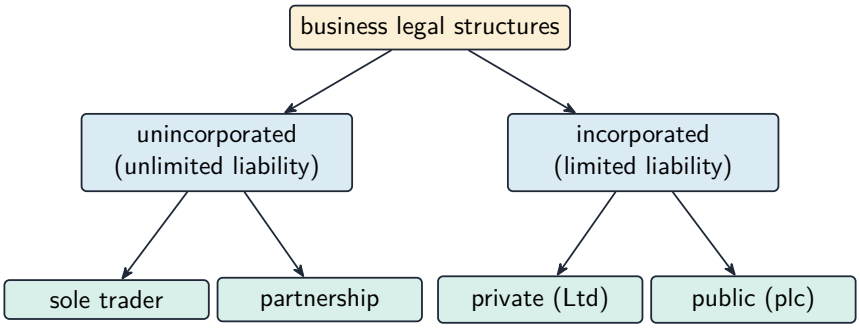


A company headquarters: businesses range from sole traders to giant public limited companies.

A business must choose a legal form. The main ones are below.

- A **sole trader** is owned by one person. It is easy to set up and the owner keeps all the profit, but raising money is hard.

- A **partnership** is owned by two or more partners who share the work and profit. They often sign a **deed of partnership** that sets out each partner's share.
- A **private limited company** (Ltd) sells **shares** to a small group, such as family and friends. Its owners are **shareholders**. Its shares are not sold to the public.
- A **public limited company** (plc) sells its shares to the public on the **stock exchange**. It can raise large amounts of money, but it must follow more rules and faces the risk of a **takeover**.
- A **franchise** lets one firm use another firm's brand and business model. The **franchisor** owns the brand; the **franchisee** pays a fee and a **royalty** to run a branch.
- A **joint venture** is when two businesses join to run one project and share the costs and risks.
- A **social enterprise** trades like a normal business but aims to do social or **environmental** good, and reinvests most of its profit.



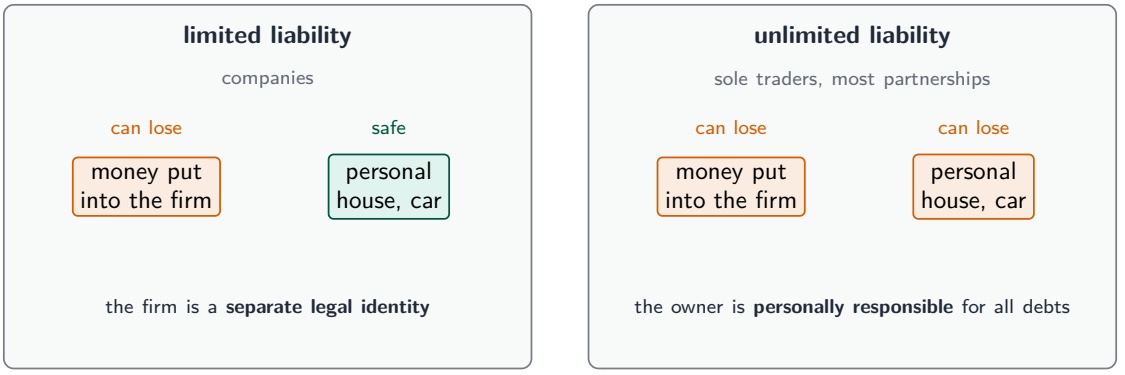
Legal forms split by liability: unincorporated (unlimited) and incorporated (limited)

Limited and unlimited liability

Liability means who must pay the business's **debts** if it cannot pay them itself.

- **limited liability** – the owners can only lose the money they put in. Their personal things, like a house or car, are safe. Companies have limited liability.
- **unlimited liability** – the owners are personally responsible for all the debts. They could lose personal things. Sole traders and most partnerships have unlimited liability.

A company is a **separate legal identity** – it can own things, owe money, and be taken to court in its own name, apart from its owners.



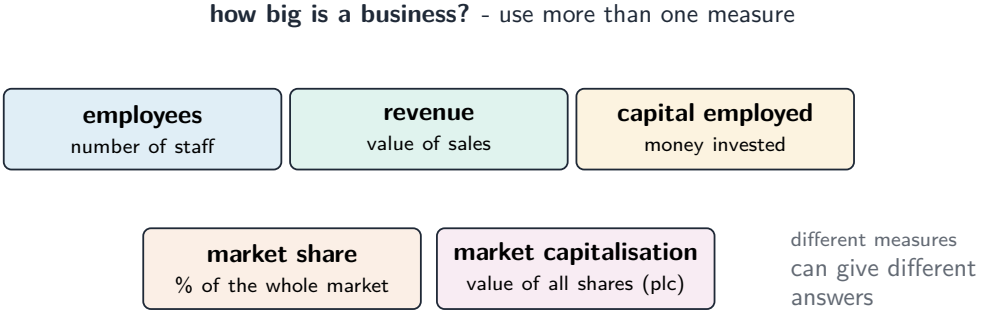
Limited liability risks only the money put in; unlimited liability risks the owner's personal things too

The size of a business

There is no single best way to measure how big a business is. Common measures are:

- the number of **employees** .
- **revenue** (also called sales turnover) – the value of what it sells.
- **capital employed** – the total money invested in the business.
- **market share** – the firm's sales as a percentage of the whole market.
- **market capitalisation** – the total value of a company's shares (for a plc).

Different measures can give different answers, so it is best to use more than one.



Five ways to measure how big a business is; use more than one

■ Why small businesses matter

Small businesses create jobs, serve local **communities** , give personal service, react quickly to change, bring new ideas, and supply parts and services to large firms.

Business growth

A business can grow in two ways.

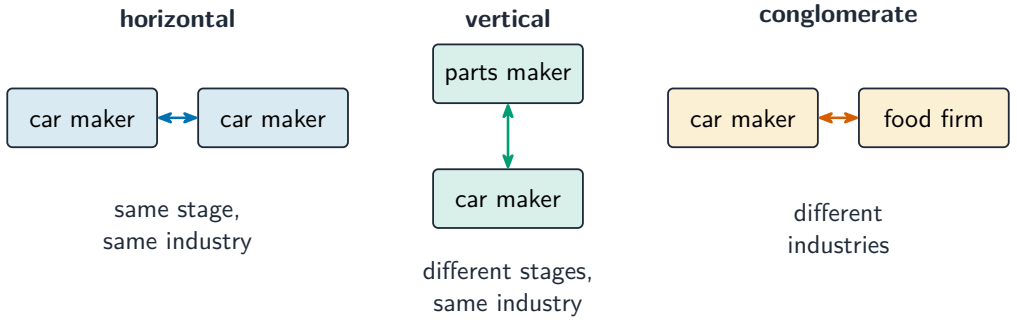
Internal growth (organic growth) – the business grows by itself: selling more, opening new branches, or adding new products. It is slower but lower risk.

External growth – the business joins with another firm. This is called **integration**, and it happens by:

- a **merger** – two firms agree to join together as one.
- a **takeover** – one firm buys control of another.

There are three types of integration:

- **horizontal integration** – joining a firm at the same stage in the same industry (e.g. two car makers).
- **vertical integration** – joining a firm at a different stage in the same industry. **Forward integration** is towards the customer (e.g. buying a shop); **backward integration** is towards the supplier (e.g. buying a parts maker).
- **conglomerate integration** – joining a firm in a completely different industry. This spreads risk, which is called **diversification**.



The three types of integration when one firm joins with another

Firms grow to lower their costs, win a bigger market share and more **market power**, raise profit, and spread risk. But growth also brings problems: higher average costs, harder management and communication, different company **cultures** that may clash, and a need for a lot of finance.

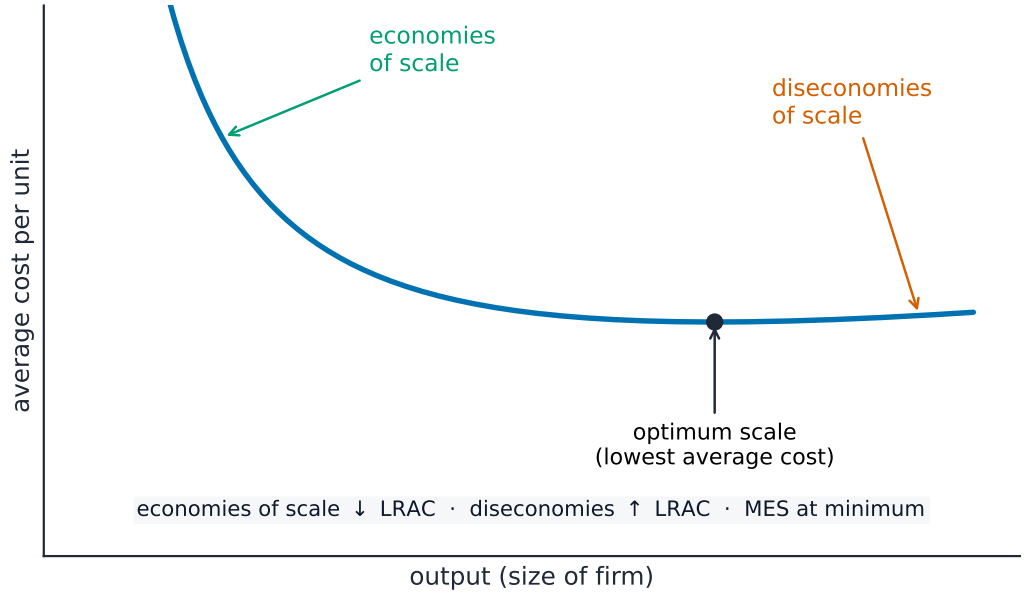
Economies and diseconomies of scale

Economies of scale are the cost savings a business gets as it grows larger. The **average cost** of each unit falls as output rises. The main types are:

Type	How it lowers cost
purchasing	buying in **bulk** earns discounts
technical	large machines are more efficient
financial	large firms borrow at lower **interest**
managerial	the firm can employ expert specialist managers
marketing	advertising cost is spread over more units

Diseconomies of scale happen when a firm grows too big and the average cost starts to rise again. The usual causes are:

- **communication** becomes slow and unclear.
- the **coordination** of many departments is hard.
- workers may feel unimportant and lose **motivation** .



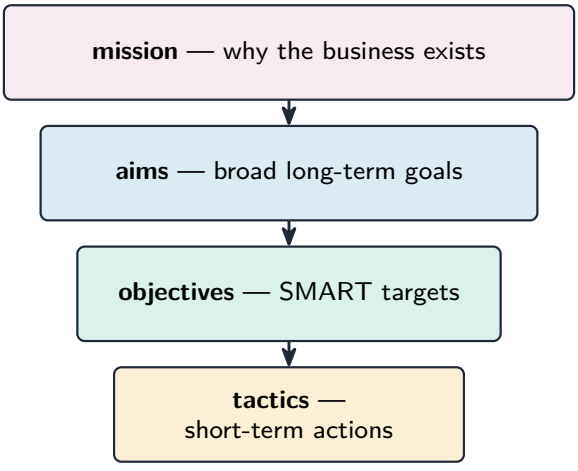
Average cost per unit falls (economies of scale) then rises (diseconomies of scale) as a firm grows

Worked example. A factory making 10,000 units has fixed costs of £200,000 and variable costs of £6 per unit. It expands to 50,000 units, and bulk buying cuts the variable cost to £5. Find the unit cost at each output and explain the fall. Unit cost = total cost ÷ output. At 10,000: total = 200,000 + (10,000 × 6) = £260,000, so the unit cost is 260,000 ÷ 10,000 = **£26**. At 50,000: total = 200,000 + (50,000 × 5) = £450,000, so the unit cost is 450,000 ÷ 50,000 = **£9**. Two economies of scale are at work: the **fixed** costs are spread over far more units (£20 down to £4 each), and **purchasing** economies cut the variable cost. Note that the fixed cost **total does not change** - only the fixed cost **per unit** falls. Writing "fixed costs fall" is the error that loses the mark.

Aims, objectives and mission

A **mission statement** is a short sentence that says why the business exists and what it wants to be. It guides everyone in the firm.

From the mission come **aims** (broad, long-term goals), and then **objectives** (the specific targets that help reach the aims).



Goals get narrower and more specific down the chain: a broad mission becomes aims, then SMART objectives, then day-to-day tactics

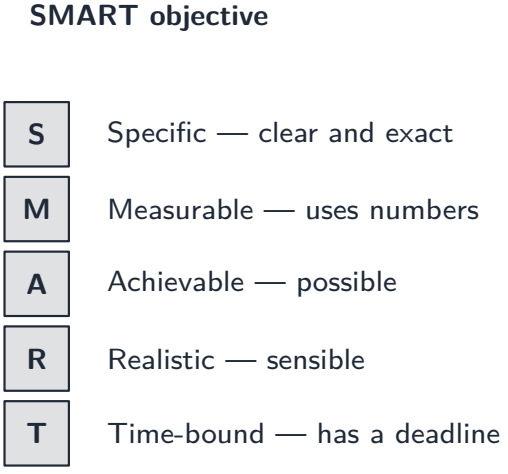
Good objectives are **SMART** – the first letters of:

- **Specific** – clear and exact.
- **Measurable** – you can check it with numbers.
- **Achievable** – it is possible to reach.
- **Realistic** – sensible for the resources you have.
- **Time-bound** – it has a deadline.

A **strategy** is the long-term plan to meet objectives. **Tactics** are the short-term actions that carry out the strategy. Objectives are shaped by the size and age of the firm, who owns it, the company culture, and conditions in the market.

Types of business objective

Objective	What it means
profit maximisation	making the largest possible profit
growth	getting bigger – more sales, more branches
survival	staying in business, vital for new firms and in hard times
market share	winning a larger share of the market
social and **ethical** objectives	doing what is morally right, such as fair pay



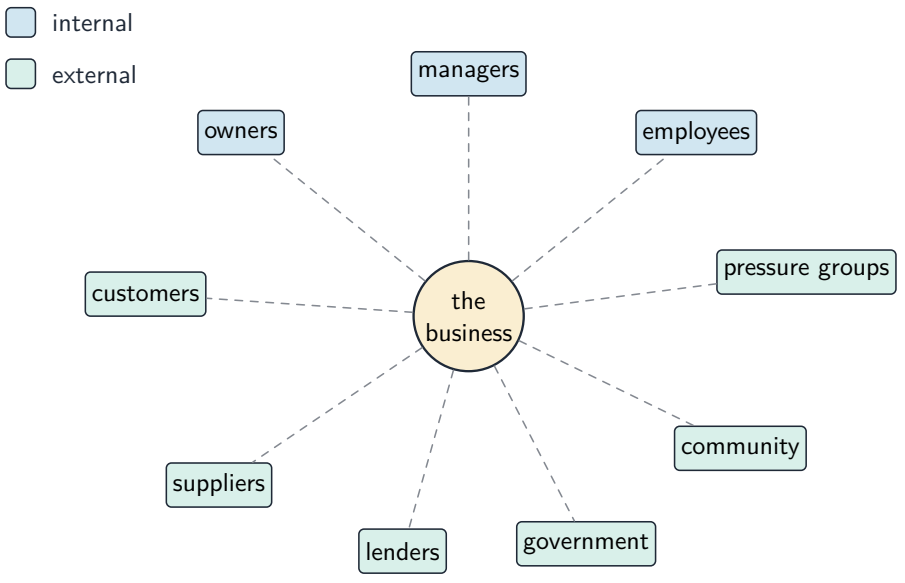
SMART objectives: Specific, Measurable, Achievable, Realistic, Time-bound

Corporate social responsibility (CSR) is the idea that a business should act in the best interests of society and the environment, not just make profit. Examples are treating workers fairly, cutting **pollution**, and helping the local community.

A firm's objectives can change over time. A new firm may aim to survive; later it may aim to grow; a large, safe firm may focus on profit or CSR.

Stakeholders

A **stakeholder** is any person or group with an interest in a business and how well it does.



Internal and external stakeholders all have an interest in the business

Internal stakeholders are inside the business:

- **owners** and shareholders – want profit and a good return.
- **managers** – want growth, job security and bonuses.
- employees – want fair pay, safe work and job security.

External stakeholders are outside the business:

Stakeholder	Main objective
customers	good quality at a fair price
suppliers	regular orders, paid on time
lenders (banks)	to be repaid with interest
the government	tax, jobs, and that the firm obeys the law
local community	jobs, but little pollution or noise
pressure groups	to protect a cause, such as the environment

■ Why stakeholder objectives conflict

Stakeholders often want different things, so their objectives can **conflict** (clash). For example:

- owners want higher profit, but employees want higher pay – and higher pay lowers profit.
- customers want low prices, but owners want a high profit margin.
- the local community wants less pollution, but cutting pollution can raise costs.

How conflict can be managed

- keep good communication with every group.
- **compromise** – give each group part of what it wants.
- **negotiation** – talk until the groups reach an agreement.
- decide which stakeholders matter most for each decision, and act on that.

Exam tips

- Define **added value** = selling price − cost of bought-in inputs, and give a way to raise it (branding, quality, design).
- Distinguish **unlimited vs limited liability** and link it to sole trader/partnership versus Ltd/plc.
- Measure business **size** in more than one way (employees, revenue, capital employed, market share) – note each has limits.
- Distinguish **objectives** (SMART, specific) from a **mission** (overall purpose).
- In evaluation, use **stakeholder conflicts** (owners vs employees vs community) to argue both sides.

Vocabulary card meanings

needs: Basic requirements a person must meet, such as food and shelter. determined: Unwilling to give up until a difficult aim is reached. self-confident: Sure of one's own ability to succeed. business start-ups: New firms in their first stage of trading. GDP: The money value of final goods and services produced inside a country in one year, so intermediate goods are not added a second time. quaternary sector: The part of the economy that sells knowledge and information services. manufacture: To make goods in quantity, usually in a factory. deed of partnership: The written agreement that sets out how partners share a firm. public limited company: A company allowed to offer shares to the public, which need not be listed on an exchange. The paired Chinese term names only a listed firm. franchise: A licence to operate an outlet using another firm's name and methods. franchisor: The owner of a brand who sells others the right to use it. franchisee: The independent owner who pays a parent firm for the right to run one outlet under that firm's brand. royalty: A fee paid for the right to use a brand or a business format. environmental: Concerned with the natural surroundings in which a firm operates. liability: What a person or a firm can be forced to pay if things go wrong. separate legal identity: The company counts as its own person in law, distinct from its owners. communities: The people who live or work around a firm and are affected by it. forward integration: A firm taking over a business closer to the final customer. backward integration: A firm taking over a supplier that sits earlier in its chain. company culture: The shared habits and values inside a firm. bulk: A large quantity bought or produced at one time. realistic: Able to be achieved with the resources the firm actually has. corporate social responsibility: A firm's duties to people and the environment beyond making profit. internal stakeholders: People

inside the firm, such as owners and staff, who are affected by its decisions. external stakeholders: People outside the firm, such as customers and neighbours, who are affected by it. lenders: People or banks that provide borrowed money.

Human resource management

A-Level Business

What human resource management does

Human resource management (HRM) is the part of a business that looks after its people. Good HRM makes sure the firm has enough staff, with the right skills, who are well trained and motivated.



Managers plan and organise the workforce – the core of human resource management.

The main roles of HRM are:

- **workforce planning** – working out how many staff, and what skills, the business will need in the future.
- **recruitment** and **selection** – finding people to apply for jobs, then choosing the best.
- **training** and **development** – improving staff skills now and helping them grow for the future.

Workforce planning, recruitment and selection

Workforce planning looks at the firm's plans, then at the staff it already has, to find the gap. The gap may be filled by hiring, training, or moving staff.



Recruitment: describe the job, advertise, then select the best applicant

To recruit, the firm first writes:

- a **job description** – the title, duties and tasks of the job.
- a **person specification** – the skills, **qualifications** and qualities the right person should have.

The firm can recruit from inside or outside:

- **internal recruitment** – filling the job with someone who already works there. It is cheaper and faster, and the person is known, but no new ideas come in.
- **external recruitment** – hiring someone from outside. It brings new skills and ideas, but costs more and is slower.

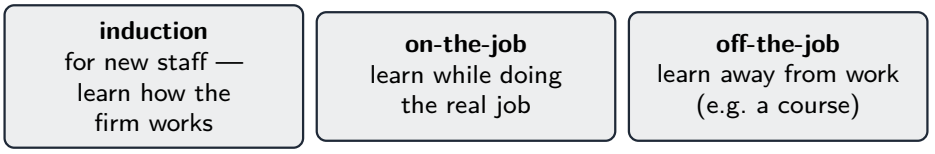
Selection methods include reading application forms and CVs, **interviews**, tests, and group tasks.



Selection: the interview is the stage most candidates meet, and the one most often assessed in the exam

Training and development

Training improves the skills staff need to do their jobs well. Common types are:



Three kinds of training: induction, on-the-job and off-the-job

- **induction training** – given to new staff so they learn how the firm works.
- **on-the-job training** – learning while doing the real job.
- **off-the-job training** – learning away from the workplace, such as a course.

Training costs time and money, but it raises quality, **productivity** and motivation, and helps keep staff.

Ending employment

Sometimes a worker has to leave.

- **redundancy** happens when the job itself is no longer needed – for example, when a machine replaces it. It is not the worker's fault.
- **dismissal** happens when a worker is made to leave because of poor work or bad behaviour.

Rights and managing change

Both sides have rights and duties. The **employer** must follow the law on pay, safety, working hours and unfair dismissal. The **employee** must work as agreed and follow safe, fair rules. These **rights** protect both sides.

Businesses change often – new technology, new markets, or **restructuring** . The **management of change** in HR means planning the change, explaining it clearly, training staff for new roles, and listening to worries, so staff accept the change instead of resisting it.

Measuring the workforce

Two simple measures help managers judge how the workforce is doing.

Labour turnover shows the percentage of staff who leave in a year:

$$\text{labour turnover} = \frac{\text{number of staff leaving}}{\text{average number employed}} \times 100\%$$

High labour turnover can mean low pay, poor motivation or weak management. It raises the cost of recruiting and training.

Worked example. A firm employs 200 people on average, and 30 leave during the year. Find its labour turnover.

$$\text{labour turnover} = \frac{30}{200} \times 100\% = 15\%.$$

Labour productivity shows the output made by each worker:

$$\text{labour productivity} = \frac{\text{total output}}{\text{number of employees}}$$

Higher labour productivity lowers the cost of each unit, so the firm can compete better.

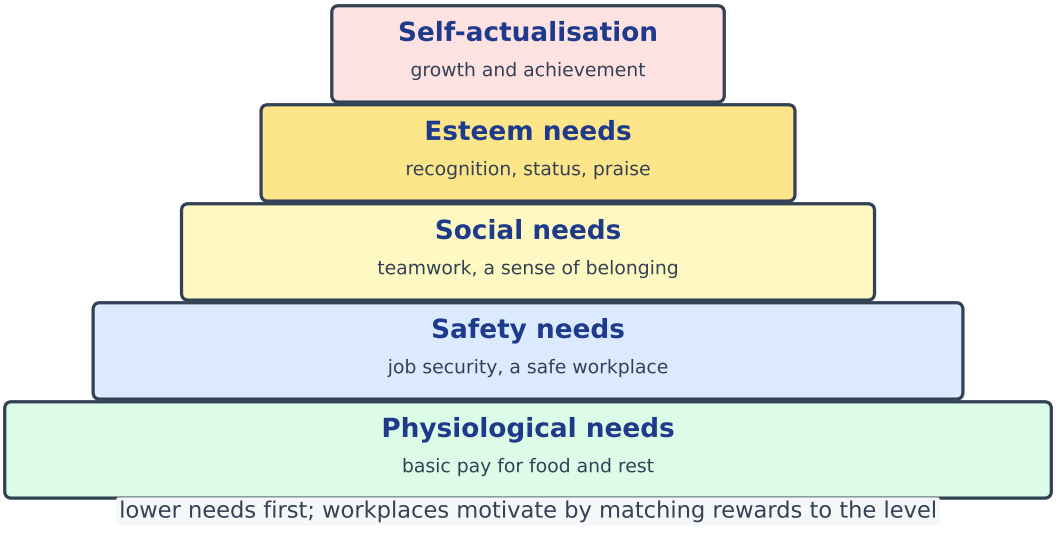
Worked example. A team of 8 workers makes 2400 units in a week. Find the labour productivity.

$$\text{labour productivity} = \frac{2400}{8} = 300 \text{ units per worker per week.}$$

Why motivation matters

Motivation is the desire to work hard and do a good job. Motivated staff produce more and better work, take fewer days off, stay longer, and give better service. So motivation lowers costs and raises quality.

Theories of motivation



Maslow's hierarchy of needs, with workplace examples at each level.

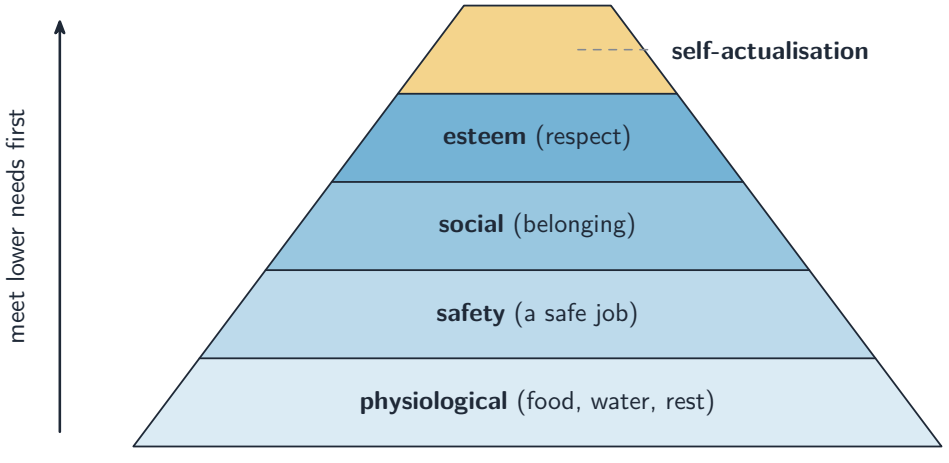
Several thinkers tried to explain what motivates people. You should know the main idea of each.

Thinker	Main idea
Taylor	**scientific management** – workers are mainly motivated by money. Pay them per unit made (**piece rate**) and they work harder.
Mayo	**human relations** – workers are also motivated by attention, teamwork and feeling part of a group.
Maslow	**hierarchy of needs** – people meet needs in order, from low to high (see below).
Herzberg	**two-factor theory** – some things cause unhappiness if missing, others truly motivate (see below).
McClelland	people are driven by three needs in different amounts: **achievement** , power and affiliation (belonging).
Vroom	**expectancy theory** – people work hard only if they believe the effort will lead to a reward they want.

Maslow's hierarchy of needs

Maslow said people try to meet lower needs first, then move up:

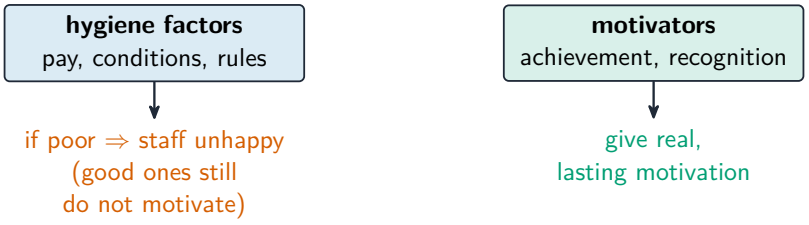
1. **physiological needs** – food, water, rest (met by pay).
2. safety needs – a safe job and workplace.
3. social needs – friendship and being part of a team.
4. **esteem needs** – respect and **recognition** .
5. **self-actualisation** – reaching your full potential.



Maslow's hierarchy: people meet lower needs before higher ones

Herzberg's two factors

- **hygiene factors** – things like pay, conditions and rules. If they are poor, staff are unhappy; but fixing them does not truly motivate.
- **motivators** – things like achievement, responsibility and recognition. These give real, lasting motivation.



Hygiene factors stop unhappiness; only motivators give real, lasting motivation

Financial motivators

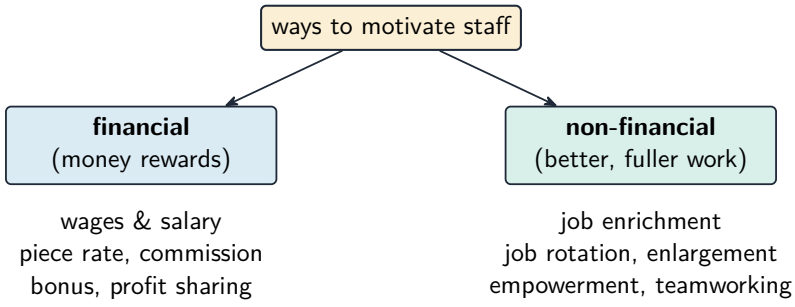
Financial motivators are rewards paid in money:

- **wages** – pay by the hour or week, often for **manual work** .
- **salary** – a fixed yearly amount, paid monthly.
- piece rate – pay for each unit made.
- **commission** – pay based on how much you sell.
- **bonus** – an extra payment for good results.
- **profit sharing** – staff get a share of the firm's profit.
- **performance-related pay** – extra pay for those who meet targets.
- **fringe benefits** – non-cash extras, such as a company car or health care.

Non-financial motivators

Non-financial motivators reward staff without extra money, by making work more interesting and giving more control:

- **job enrichment** – giving more challenging tasks and responsibility.
- **job rotation** – moving between different tasks to reduce boredom.
- **job enlargement** – adding more tasks at the same level.
- **empowerment** – letting staff make their own decisions about their work.
- **teamworking** – organising staff into teams that share goals.



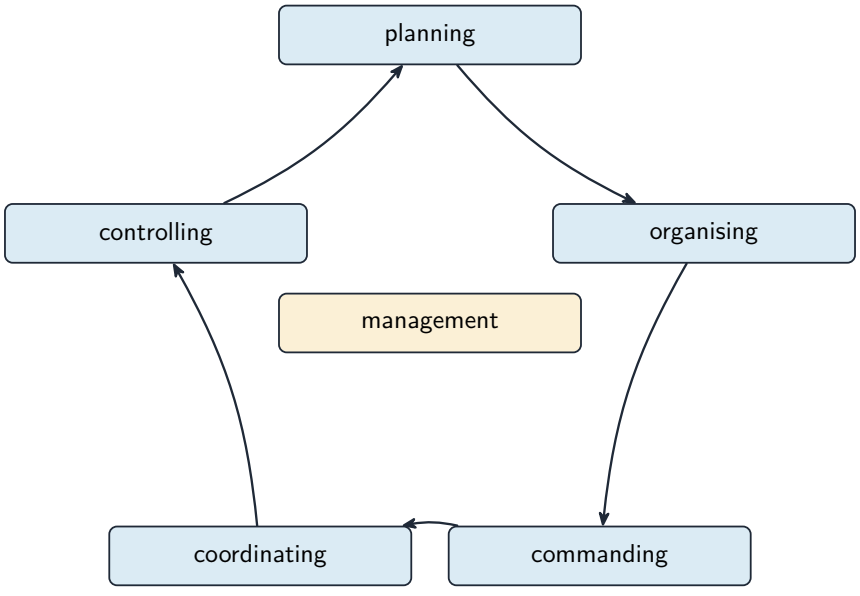
Motivators are either financial (money) or non-financial (better, fuller work)

Management and its functions

Management means getting work done through other people to reach the firm's objectives.

Fayol said managers carry out five functions:

Function	What the manager does
planning	set objectives and decide how to reach them
organising	arrange people and resources
commanding	give clear instructions and guidance
coordinating	make sure all parts work together
controlling	check results against the plan and correct problems



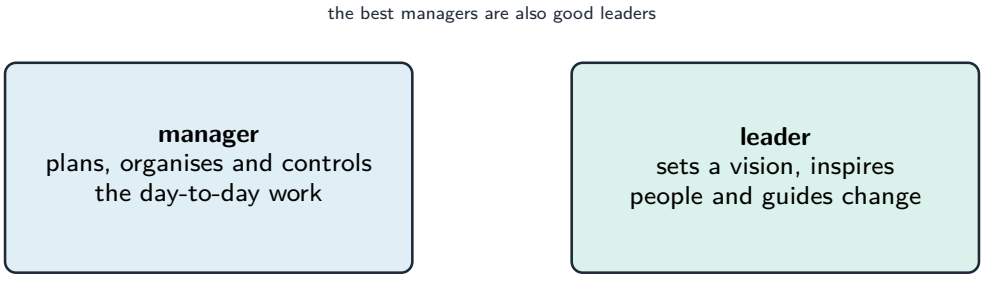
Fayol's five functions of management form a repeating cycle

Mintzberg said managers also play ten **roles** in three groups: interpersonal (working with people), informational (sharing information), and decisional (making decisions). This shows that real management is busy and varied, not just the five neat functions.

Leadership versus management

Leadership and management are not the same.

- a manager plans, organises and controls the day-to-day work.
- a leader sets a **vision**, inspires people, and guides change.



A manager runs the day-to-day work; a leader sets direction and inspires – the best people do both

The best managers are also good leaders. Effective management matters because it raises productivity, keeps good staff, helps the firm adapt to change, and turns plans into real results.

Exam tips

- Calculate and interpret **labour turnover** and **labour productivity**; high turnover raises recruitment and training costs.
- Link a **motivation theory** to a method: Maslow (needs), Herzberg (hygiene vs motivators), Taylor (pay).
- Distinguish **financial** (piece rate, bonus) from **non-financial** (job enrichment, empowerment) motivators.
- Distinguish **leadership from management** and match a style (autocratic, democratic, laissez-faire) to the situation.

Vocabulary card meanings

development: Helping staff gain skills, rather than inventing a new product. dismissal: Ending someone's job because of their conduct or performance. employer: The person or firm that hires and pays staff. employee: A person who works for a firm in return for pay. rights: Entitlements that the law or a contract gives a person at work. restructuring: Changing how a firm is organised, often to cut costs. management of

change: The way leaders plan and support people through a new way of working. hierarchy of needs: Maslow's order of human requirements, from survival up to personal growth. two-factor theory: Herzberg's split between things that prevent dissatisfaction and things that motivate. achievement: The satisfaction of reaching a demanding goal. expectancy theory: The idea that effort depends on whether people expect it to lead to a valued reward. physiological needs: The basic bodily requirements, such as food, water and rest. esteem needs: The desire to be respected and to respect oneself. financial motivators: Rewards paid in money, such as wages, bonuses or commission. manual work: Jobs done mainly with the body rather than at a desk. non-financial motivators: Rewards other than money, such as praise, responsibility or training.

Marketing

A-Level Business

The role of marketing

Marketing is the job of finding out what customers want, then making and selling products that meet those wants at a profit. Good marketing links the business to its **market** (all the buyers and sellers of a product).

A business can be:

- **product-orientated** – it focuses on the product first and hopes customers will buy it. This suits new or technical products.
- **market-orientated** – it finds out what customers want first, then makes it. This lowers the risk of failure.

Customer relationship marketing (CRM) means building long-term links with customers so they come back again. Keeping a customer is cheaper than finding a new one.

Market size, growth and share

- **market size** – the total sales of all firms in a market, by value or by number of units.
- **market growth** – how fast the market is getting bigger, shown as a percentage.
- **market share** – one firm's sales as a percentage of the total market:

$$\text{market share} = \frac{\text{firm's sales}}{\text{total market sales}} \times 100\%$$

A rising market share is a sign of strong marketing.

> **Worked example.** A firm sells £4 million of goods in a market worth £50 million in total. Find its market share.
>
> $$\text{market share} = \frac{4}{50} \times 100\% = 8\%.$$

If the whole market grows but the firm's sales stay at £4 million, its market share **falls** – so a firm can sell the same amount yet lose ground to faster-growing rivals.

Mass and niche markets

- a **mass market** is large, with products aimed at most buyers (e.g. soft drinks). Sales are high, but competition is strong.
- a **niche market** is a small part of a market with special needs (e.g. left-handed tools). Competition is weaker, but sales are smaller.



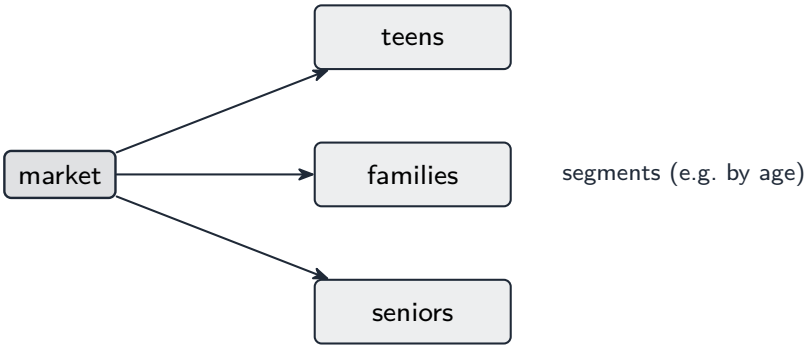
A mass market targets everyone; a niche serves a small specialist group

Market segmentation



Shelf space is a marketing decision: segment, position, then price and promote

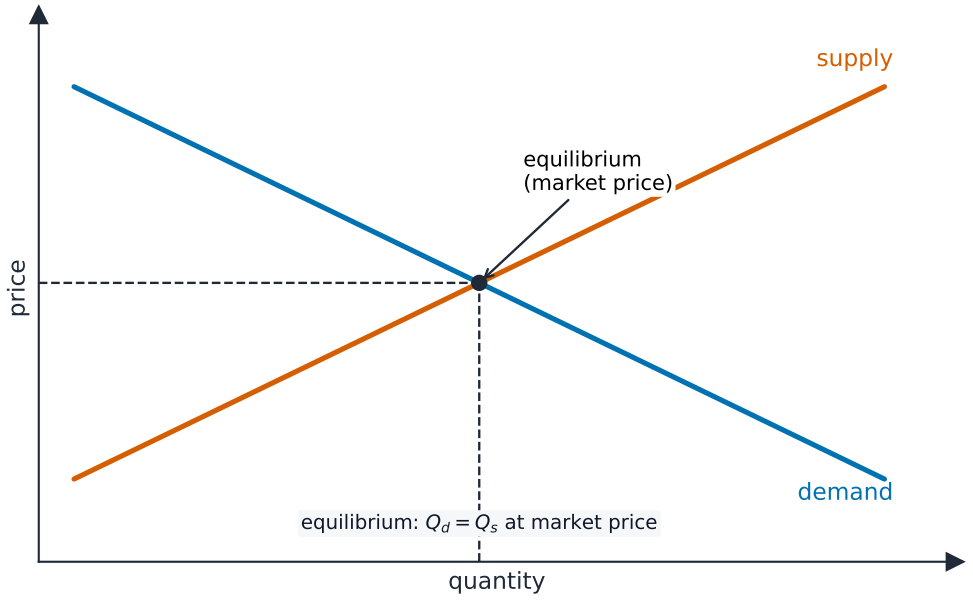
Market segmentation means splitting a market into groups (**segments**) of buyers who are alike. Common ways to segment are by age, income, gender, location, and lifestyle. Segmentation lets a firm aim the right product and message at each group, which raises sales and cuts waste.



Splitting a market into segments by age, income and lifestyle

Demand, supply and elasticity

Demand is how much of a product buyers will buy at each price. **Supply** is how much sellers will offer at each price. Usually, when price falls, demand rises.



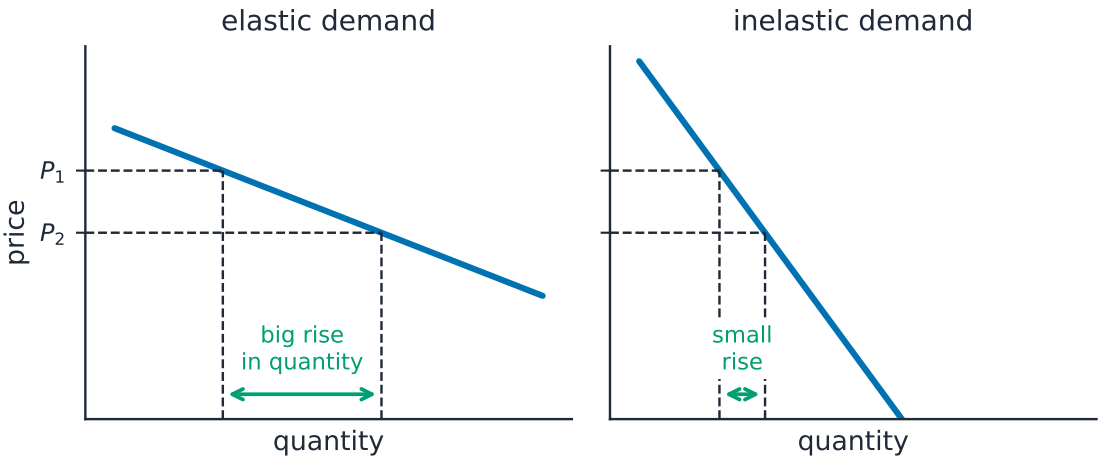
Supply and demand meet at the equilibrium market price

Elasticity measures how strongly demand reacts to a change. You should know three types.

Price elasticity of demand (PED) measures how demand reacts to a price change:

$$\text{PED} = \frac{\%\text{ change in quantity demanded}}{\%\text{ change in price}}$$

- if PED is greater than 1, demand is **price elastic** – a small price cut raises sales a lot, so lowering price can raise revenue.
- if PED is less than 1, demand is **price inelastic** – demand changes little, so raising price can raise revenue.



$$\text{PED} = \frac{\%\,\Delta Q_d}{\%\,\Delta P} \cdot |PED| > 1 \text{ elastic} \cdot |PED| < 1 \text{ inelastic}$$

The same price fall raises quantity a lot when demand is elastic, only a little when inelastic

Income elasticity of demand (YED) measures how demand reacts to a change in customers' income. Luxury goods react strongly; basic goods react weakly.

Cross elasticity of demand (XED) measures how the demand for one product reacts to a price change of another. It is positive for **substitutes** (e.g. tea and coffee) and negative for **complements** (e.g. cars and fuel).

Elasticity is useful because it helps a firm predict how a price change will affect sales and revenue.

Market research

Market research is collecting and studying information about customers, **competitors** and the market.

- **primary research** – new information you collect yourself, by surveys, interviews or observation. It fits your exact needs, but costs more and takes time.
- **secondary research** – information that already exists, such as reports, websites and sales records. It is cheap and quick, but may be old or not exactly right.

Data can be:

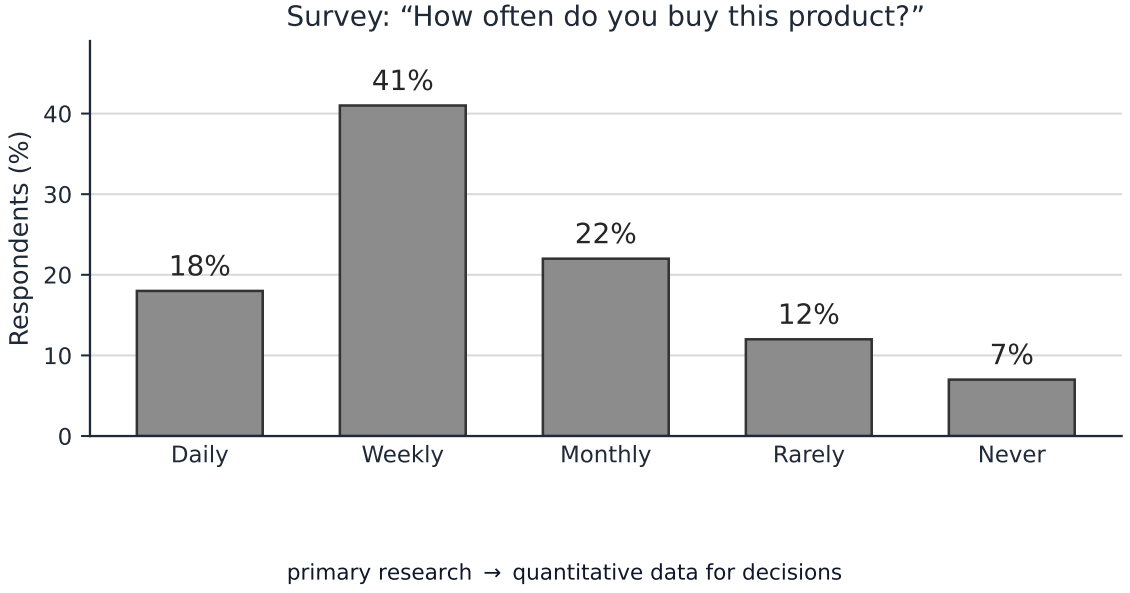
- **quantitative data** – facts in numbers (e.g. "60% buy weekly"). Easy to compare.
- **qualitative data** – opinions and reasons (e.g. "why people like the brand"). Gives deeper understanding.

■ Sampling and reliability

A **sample** is the small group of people asked, chosen to stand for the whole market. **Sampling** methods include:

- **random sampling** – everyone has an equal chance of being picked.
- **quota sampling** – set numbers from each group are asked.
- **stratified sampling** – the sample copies the make-up of the whole market.

A larger, well-chosen sample gives more **reliable** results. Poor sampling can cause **bias**, which makes the findings wrong. Results are often shown in tables and charts, then read carefully to guide decisions.



Primary research → quantitative data that guides marketing decisions

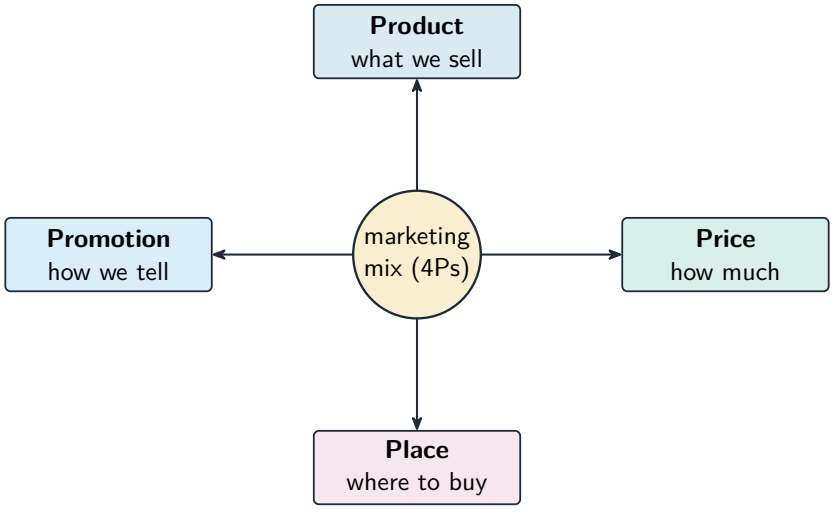
The marketing mix



A branded shop window – branding and place help sell the product.

The **marketing mix** is the set of choices a firm makes to sell a product. It is often called the **4Ps**:

P	Question it answers
product	what are we selling?
price	how much do we charge?
place	where and how do customers buy it?
promotion	how do we tell customers about it?



The marketing mix combines the four Ps

The four parts must work together. For example, a high price needs high quality and a good image. Making the parts fit each other is called **integration** .

Product

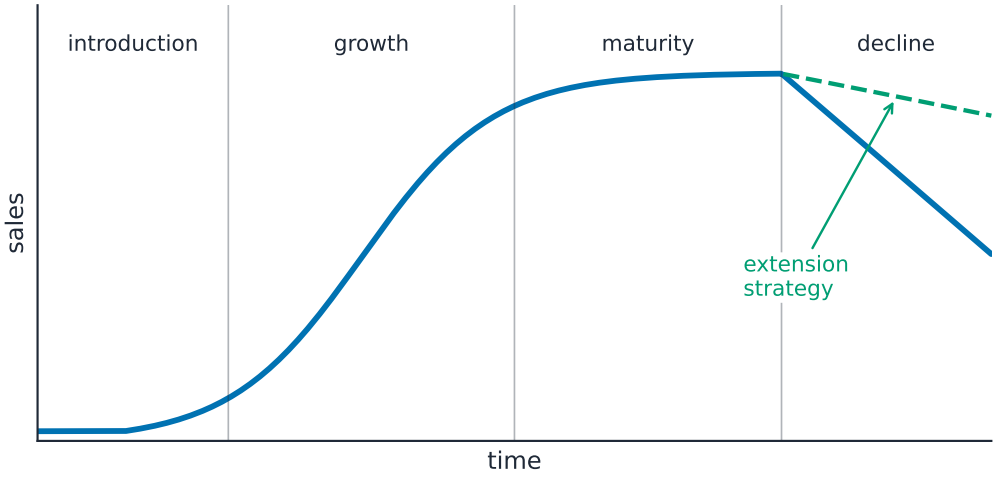
A good product is the heart of the mix.

Product development is the work of designing and improving products to meet customer needs and beat rivals.

■ The product life cycle

The **product life cycle** shows the sales of a product over time, in four stages:

1. **introduction** – launch; low sales, high promotion cost.
2. **growth** – sales rise quickly.
3. **maturity** – sales reach their peak and level off.
4. **decline** – sales fall as tastes change.



introduction → growth → maturity → decline (sales over time)

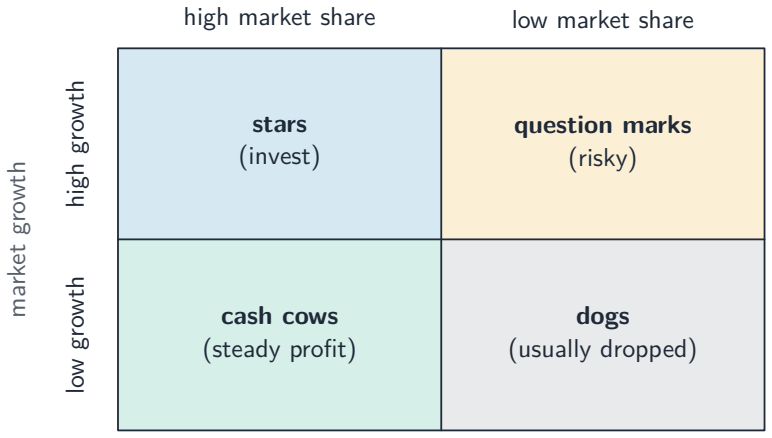
Sales of a product over its four life-cycle stages, with an extension strategy in maturity

To keep sales up in maturity, firms use **extension strategies**, such as new packaging, new uses, or selling in new markets.

■ The product portfolio (Boston Matrix)

The **product portfolio** is the full range of products a firm sells. The **Boston Matrix** sorts products by market share and market growth:

	High market growth	Low market growth
High share	**stars**	**cash cows**
Low share	**question marks**	**dogs**



The Boston Matrix sorts products by market share and market growth

Stars need investment but may become cash cows. Cash cows bring steady profit. Question marks are risky. Dogs are usually dropped.

■ Branding

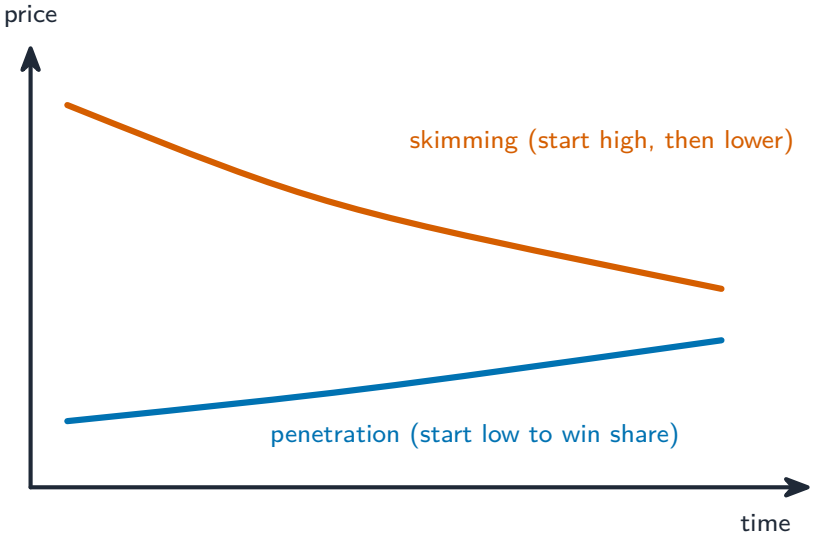
Branding gives a product a name, design and image that customers recognise. A strong brand builds **loyalty**, lets the firm charge more, and makes new products easier to launch.

Price

Choosing a price is a key decision. The main methods and strategies are:

Method / strategy	How it works
cost-plus pricing	add a fixed profit margin to the unit cost
penetration pricing	set a low price to enter a market and win share
price skimming	set a high price first for a new, special product, then lower it
competitive pricing	set a price close to rivals' prices
price discrimination	charge different prices to different groups (e.g. peak vs off-peak)
dynamic pricing	change the price often as demand changes
psychological pricing	use prices like \$9.99 that feel lower

The best price depends on costs, competitors, the product's life-cycle stage, and how price elastic demand is.



Two launch strategies over time: penetration sets a low price to win share; skimming sets a high price first, then lowers it

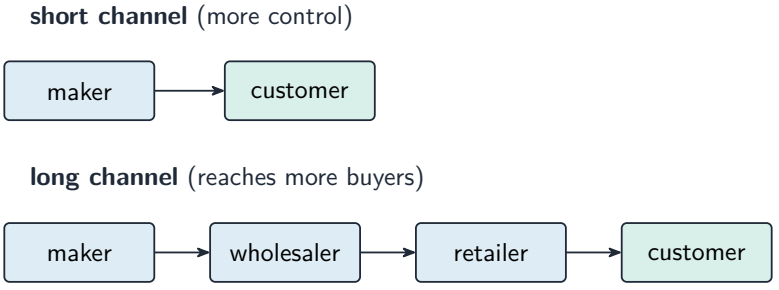
Place and promotion



Advertising on giant screens – promotion is part of the marketing mix.

Place is how the product reaches the customer, through a **distribution channel**

. A short channel (maker → customer) gives more control; a longer channel (maker → **wholesaler** → shop → customer) reaches more buyers.



A short channel gives more control; a longer channel reaches more buyers

Promotion tells customers about a product and persuades them to buy. It is split into:

- **above-the-line promotion** – paid **advertising** in mass media, such as TV and online ads, aimed at a wide audience.
- **below-the-line promotion** – targeted methods the firm controls, such as discounts, free samples and direct mail.

Digital marketing and **e-commerce** (buying and selling online) now play a huge role. They let small firms reach world markets cheaply, target adverts to the right people, and sell at any time of day.

Exam tips

- Calculate **market share** and **market growth** as percentages; a firm can grow yet lose share in a faster-growing market.
- Use **price elasticity of demand** to decide pricing: raise price when demand is inelastic (revenue rises), cut it when elastic.
- Distinguish **primary vs secondary** and **quantitative vs qualitative** research, and comment on sampling reliability.
- Treat the **marketing mix (4Ps)** as a coordinated set – each P should support the others.

Vocabulary card meanings

customer relationship marketing: Building long-term links with buyers rather than chasing one-off sales. market segmentation: Splitting buyers into groups that share similar wants. segments: Groups of buyers who share similar wants. price elastic: Quantity demanded changes by a larger percentage than the price. price inelastic: Quantity demanded changes by a smaller percentage than the price. competitors: Other firms that sell to the same buyers. reliable: Able to be trusted to work or to tell the truth. Boston Matrix: A four-box chart that places products by their relative share

of a market and by how fast that market is growing. stars: Products with a high share of a fast-growing market. cash cows: Products with a high share of a slow-growing market. question marks: Products with a low share of a fast-growing market. dogs: Products with a low share of a slow-growing market. psychological pricing: Setting a price, such as 9.99, to influence how buyers feel about it. distribution channel: The route a product takes from the producer to the buyer. wholesaler: A firm that buys in bulk from producers and sells on to shops. advertising: Paid messages that promote a product to many people at once. digital marketing: Promotion that uses the internet, apps or social media. e-commerce: Buying and selling through the internet.

Operations management

A-Level Business

What operations management does

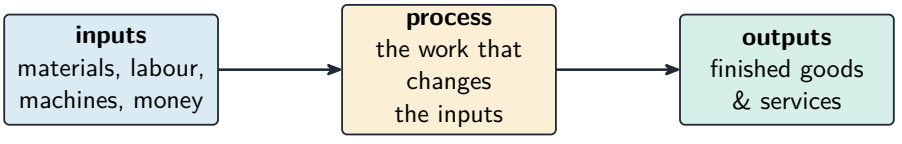
Operations management is the part of a business that makes the product. **Operations management** is the job of turning **inputs** into **outputs** well.



A distribution centre: operations management turns inputs into finished goods ready for customers.

This is the **transformation process**:

- **inputs** – raw materials, labour, machines and money.
- **process** – the work that changes the inputs.
- **outputs** – finished goods and services for the customer.



Operations turn inputs into outputs, adding value along the way

Good operations **add value**: the output is worth more than the inputs used to make it. Using **raw materials** (the basic materials a business starts with) well is a big part of this.

Productivity and efficiency

Productivity measures how much output you get from your inputs:

$$\text{productivity} = \frac{\text{output}}{\text{input used}}$$

Efficiency means making products with as little waste of time, materials and money as possible. Higher productivity and efficiency lower the cost of each unit, so the firm can charge less or earn more.

> **Worked example.** A factory of 20 workers makes 800 units in a day. Find the output per worker.

$$\text{productivity} = \frac{800}{20} = 40 \text{ units per worker per day.}$$

If training raised output to 1000 units with the same 20 workers, productivity would rise to 50 units each – the same staff cost now spread over more units, so each unit is cheaper to make.



Assembly line: productivity is output per worker (or per hour) – rising productivity cuts unit costs

Capital- and labour-intensive operations

- a **capital-intensive** operation uses mostly machines (e.g. a car factory). It has high set-up costs but low cost per unit, and is good for large amounts.
- a **labour-intensive** operation uses mostly people (e.g. a hair salon). It is flexible and good for personal service, but quality can vary.



Factory robots: capital-intensive production uses more machines relative to labour

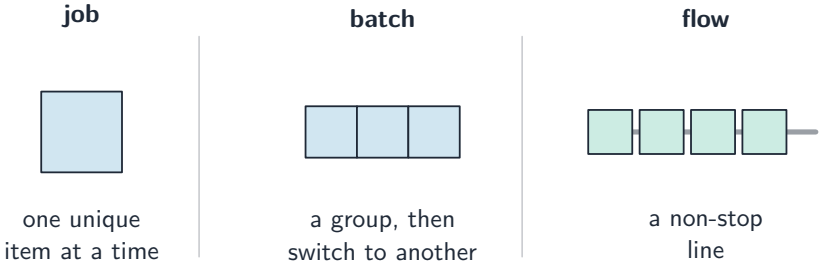
Methods of production

A firm chooses how to organise production to suit its product and demand.



A car assembly line: flow production of identical units, one stage after another

Method	What it is	Best for
job production	making one unique item at a time	a wedding cake, a bridge
batch production	making a group of the same item, then switching	bread in flavours
flow production	making items non-stop on a line	bottled drinks, cars
mass customisation	a flow line that still lets customers choose options	cars with chosen colours



The three methods of production

Job production is flexible but costly per unit. Flow production has a low unit cost but needs high, steady demand.

Sustainability and technology

Sustainability means meeting today's needs without harming the future or the planet. Sustainable operations cut waste, use less energy, recycle, and use materials that can be replaced.

Technology is widely used in operations. **Automation** (machines and robots doing tasks) raises output, improves quality and lowers cost, but it has a high set-up cost and can replace jobs.

Inventory and why it matters

Inventory (stock) is the goods a business holds: raw materials, part-finished goods, and finished goods waiting to be sold.

A firm holds inventory so it can keep producing and meet customer orders quickly. But holding inventory has costs:

- **holding costs** – storage, insurance, and money tied up in stock.

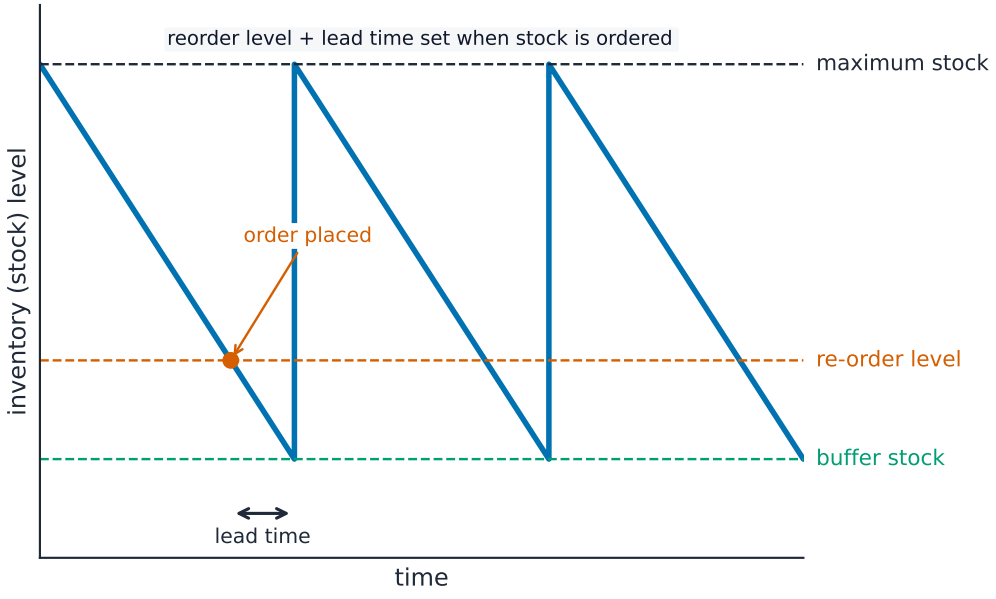
- the risk that goods go out of date or are damaged.

Holding too little stock is also risky: the firm may run out and lose sales.

Inventory control charts

An **inventory control chart** shows how stock rises when an order arrives and falls as it is used. Key terms:

- **re-order level** – the stock level at which a new order is placed.
- **lead time** – the time between placing an order and receiving it.
- **buffer inventory** – a minimum "safety" stock kept in case of delays or extra demand.
- **maximum inventory** – the most stock the firm wants to hold.

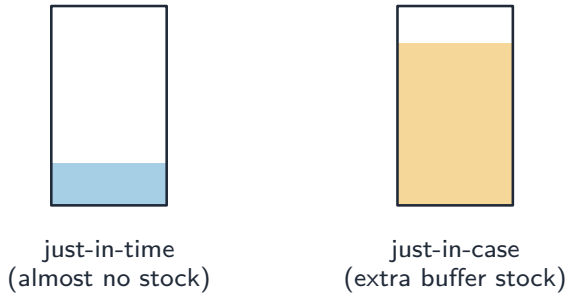


Stock is used up, then replenished when a new order arrives after the lead time

Just-in-time and just-in-case

There are two main ways to manage stock.

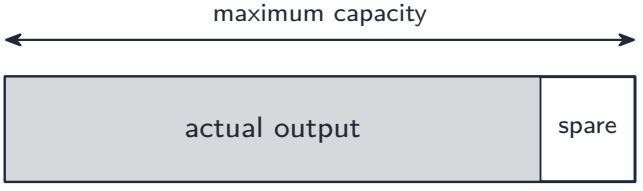
- **just-in-time** (JIT) – stock arrives only as it is needed, so almost no inventory is held. This cuts holding costs and waste, but a late delivery can stop production.
- **just-in-case** – extra stock is held in case of problems. This is safer but costs more to store.



Just-in-time holds almost no stock (low cost, but risky if a delivery is late); just-in-case keeps a buffer (safer, but more to store)

Capacity utilisation

Capacity is the most a business can produce with its current resources. **Capacity utilisation** shows how much of that capacity is being used:



Capacity utilisation: how much of the maximum output is used

$$\text{capacity utilisation} = \frac{\text{actual output}}{\text{maximum possible output}} \times 100\%$$

- **under-utilisation** (a low figure) means resources sit idle, so the fixed cost spread over each unit is high.
- **over-utilisation** (very close to 100%) leaves no time for repairs, can tire staff, and may lower quality.

Most firms aim for a high but not full level, around 85–90%.

> **Worked example.** A factory can make 5000 units a week but currently makes 4000. Find its capacity utilisation.

$$\text{capacity utilisation} = \frac{4000}{5000} \times 100\% = 80\%.$$

This leaves 20% spare. Winning an order for another 500 units a week would lift utilisation to 90%, spreading the fixed costs over more units without needing new machines.

Outsourcing

Outsourcing means paying another firm (a **supplier**) to do work the business used to do itself, such as cleaning, IT or making parts.

Benefits: it can be cheaper, lets the firm focus on what it does best, and adds flexibility. Drawbacks: less control over quality, possible delays, and the risk of sharing secrets with another firm.

Exam tips

- Calculate **capacity utilisation** and **productivity**; high utilisation spreads fixed costs but leaves no slack for breakdowns or new orders.
- Read an **inventory control chart** (re-order level, lead time, buffer stock, maximum stock).
- Compare **JIT vs JIC** (low holding cost but delivery risk vs security of supply but higher cost).
- Match the **production method** (job, batch, flow) to the product and the size of the order.

Vocabulary card meanings

transformation process: Turning inputs such as materials and labour into outputs. mass customisation: Making large numbers of goods that can still be adapted to each buyer. inventory control chart: A diagram that tracks stock against reorder level and buffer stock. capacity utilisation: Current output as a percentage of the maximum output a firm could make. under-utilisation: Using much less of the available capacity than the firm could. over-utilisation: Running above a comfortable level of capacity, which strains staff and machines.

Finance and accounting

A-Level Business

Why a business needs finance

Business finance is the money a firm needs to start, run and grow. A new firm needs money to buy **premises** and equipment; a growing firm needs money for new projects; and every firm needs money to pay day-to-day bills.



Banks are a key external source of finance, through loans and overdrafts.

Capital and revenue expenditure

Spending falls into two types:

capital expenditure
fixed assets that last
(machines, vehicles, buildings)

revenue expenditure
day-to-day running
(wages, rent, materials)

Capital spending buys fixed assets; revenue spending covers running costs

- **capital expenditure** – money spent on **fixed assets** that last a long time, such as machines, vehicles and buildings.
- **revenue expenditure** – money spent on day-to-day running, such as wages, rent and raw materials.

Working capital

Working capital is the money a firm has for its daily needs. It is found from the **balance sheet** :

$$\text{working capital} = \text{current assets} - \text{current liabilities}$$

Here **current assets** are things that become cash within a year (cash, inventory, money owed by customers), and **current liabilities** are debts due within a year. Too little working capital is dangerous – the firm may not be able to pay its bills.



Cash is working capital: firms need enough liquid assets to pay day-to-day bills

Cash is not the same as profit

This is a key idea.

- **profit** is what is left when total costs are taken from total revenue over a period.
- **cash** is the money the firm actually has right now to spend.

A firm can be **profitable** but still run out of cash – for example, if it sells goods but customers have not paid yet. Many firms fail not from low profit, but from running out of cash.

a firm can be profitable but still run out of cash

profit (on paper)
revenue − costs = positive

cash (in the bank)
low — customers
haven't paid yet

Profit and cash are different: a firm can look profitable on paper yet have no cash to pay its bills – which is why many fail

Sources of finance

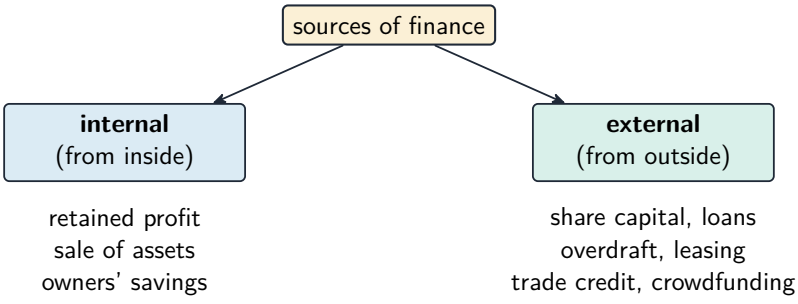


A stock exchange: public limited companies raise finance by selling shares.

Finance can come from **inside** the business (**internal sources**) or from **outside** (**external sources**). It can also be **short-term** (paid back within a year) or **long-term** .

Source	Type	What it is
retained profit	internal, long	profit kept in the business instead of paid out
sale of assets	internal	selling items the firm no longer needs
share capital	external, long	money raised by selling shares

Source	Type	What it is
loan	external, long	borrowed money repaid with interest
debenture	external, long	a long-term loan certificate sold by a company
overdraft	external, short	the bank lets the account go below zero for a short time
leasing	external	renting an asset instead of buying it
hire purchase	external	buying an asset by paying in instalments
trade credit	external, short	paying a supplier later (e.g. after 30 days)
venture capital	external, long	money from investors into risky young firms
crowdfunding	external	small amounts raised from many people online
microfinance	external	small loans for people with little access to banks



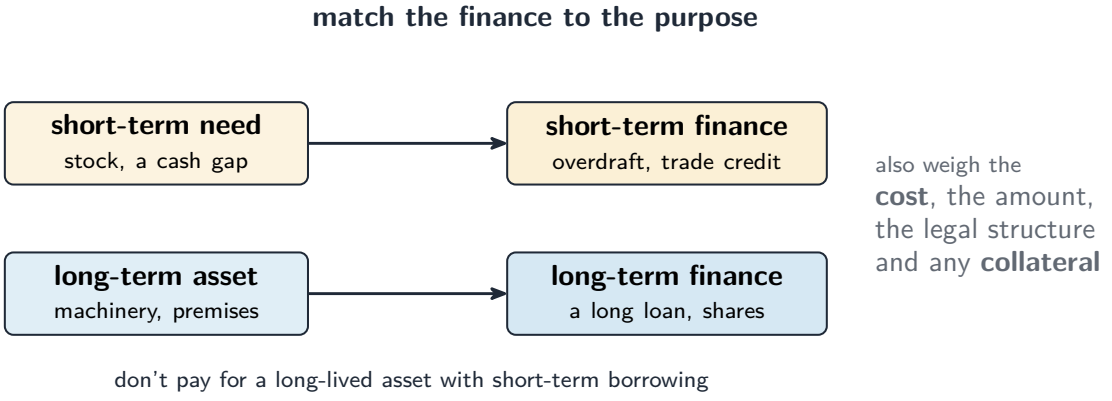
Sources of finance come from inside the business (internal) or outside it (external)

Choosing a source of finance

There is no single best source. The choice depends on:

- the **amount** needed – large amounts may need shares or a long loan.
- the **purpose** – long-term assets should use long-term finance.
- the **cost** – interest and fees differ between sources.
- the **legal structure** – only a company can sell shares.

- how much the firm already owes, and whether it can offer **collateral** (an asset the lender can take if the loan is not repaid).



Match the finance to the purpose: short-term needs use short-term finance, long-term assets use long-term finance

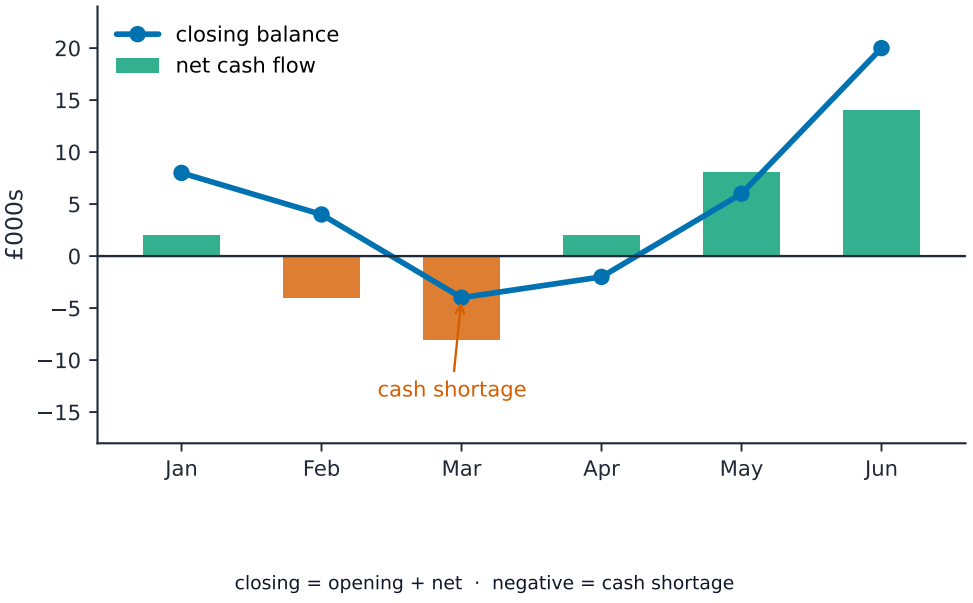
Cash-flow forecasts

A **cash-flow forecast** predicts the cash coming in and going out each month. It helps a firm spot a shortage early.

- **cash inflows** – money coming in, mostly from sales.
- **cash outflows** – money going out, such as wages and rent.
- **net cash flow** – inflows minus outflows for the period.

$$\text{net cash flow} = \text{cash inflows} - \text{cash outflows}$$

The **opening balance** is the cash at the start of the month. Adding net cash flow gives the **closing balance**, which becomes next month's opening balance.



A cash-flow forecast tracks the closing balance each month, warning of a shortage early

Cash-flow problems and how to fix them

Common causes of cash-flow problems are: too many sales on credit, holding too much stock, buying too many assets at once, and **overtrading** (growing too fast for the cash available).

Ways to improve cash flow:

- bring cash in sooner – ask customers to pay faster, or offer discounts for quick payment.
- delay cash going out – agree longer trade credit with suppliers.
- arrange an overdraft or short loan to cover a gap.
- hold less stock to free up cash.

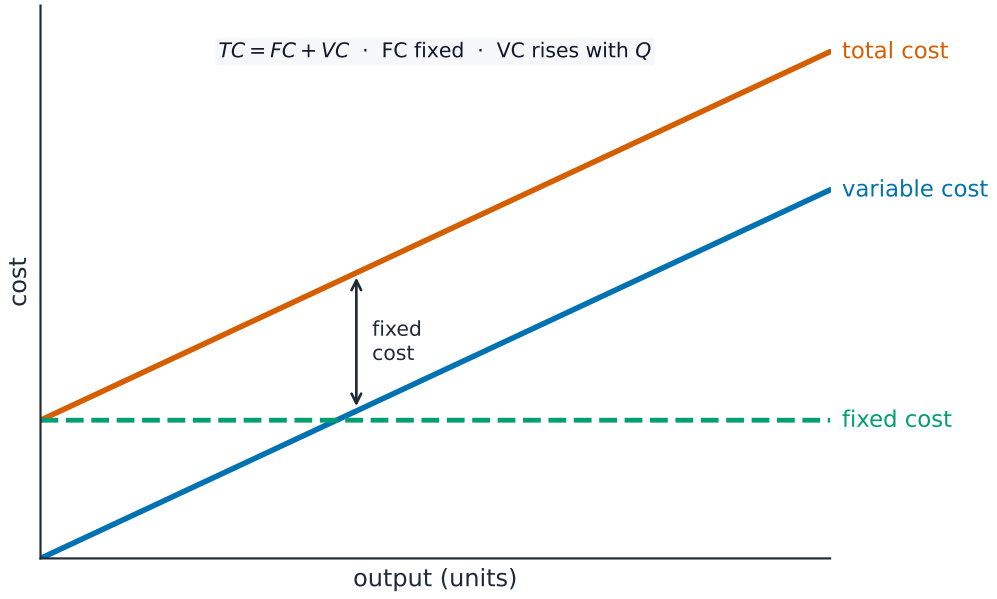
Good **working capital management** keeps enough cash to stay safe without holding so much that money sits idle.

Costs and how we classify them

A **cost** is money the business spends to make its product. We sort costs in several ways:

- **fixed costs** – costs that do not change with output, such as rent.
- **variable costs** – costs that rise and fall with output, such as raw materials.
- **direct costs** – costs clearly linked to one product, such as its materials.

- **indirect costs** (overheads) – costs not linked to one product, such as manager salaries.
- **marginal cost** – the cost of making one more unit.
- **average cost** – the cost per unit, found by total cost ÷ number of units.



Total cost = fixed cost + variable cost; only variable cost rises with output

Cost information helps managers set prices, choose what to make, and control spending.

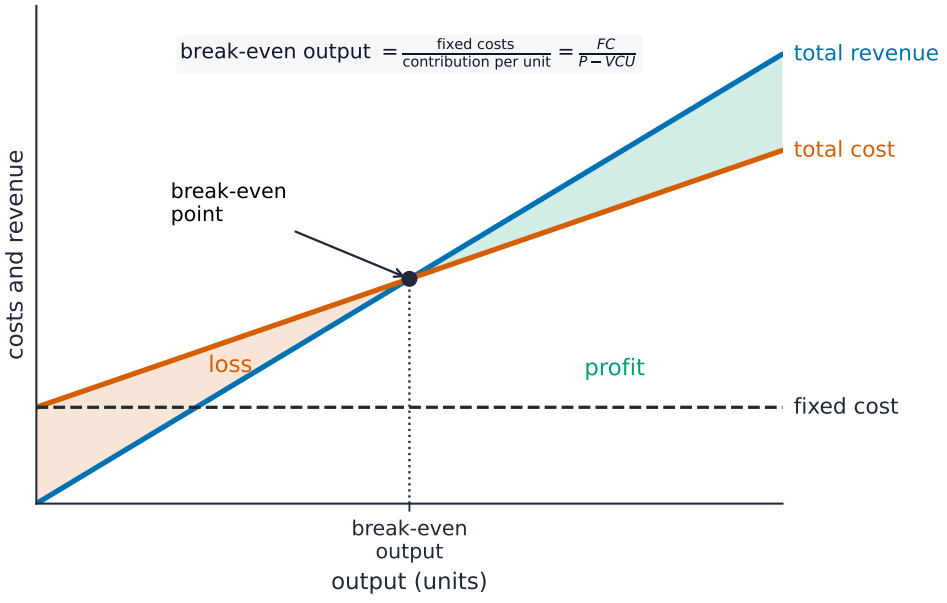
Break-even analysis

Contribution is how much each sale adds towards paying the fixed costs:

$$\text{contribution per unit} = \text{selling price} - \text{variable cost per unit}$$

Break-even is the point where total revenue equals total costs, so profit is zero. The **break-even point** in units is:

$$\text{break-even point} = \frac{\text{fixed costs}}{\text{contribution per unit}}$$



The break-even point is where total revenue equals total cost – left of it is loss, right of it is profit

The **margin of safety** is how far current sales are above the break-even point:

$$\text{margin of safety} = \text{actual output} - \text{break-even output}$$

Worked example. A product sells for £20 with a variable cost of £12 per unit. Fixed costs are £40,000, and the firm makes 6,000 units. Find the contribution per unit, the break-even output and the margin of safety.

$$\text{contribution per unit} = £20 - £12 = £8$$

$$\text{break-even point} = \frac{£40\,000}{£8} = 5\,000 \text{ units}$$

$$\text{margin of safety} = 6\,000 - 5\,000 = 1\,000 \text{ units}$$

A large margin of safety means sales can fall a lot before the firm makes a loss. Break-even analysis is quick and useful, but it has **limitations**: it assumes the selling price and costs stay the same, and that everything made is sold.

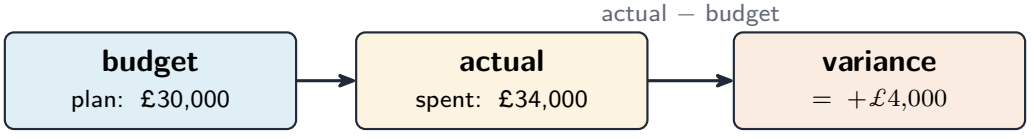
Budgets and variances

A **budget** is a financial plan for the future – a target for income or spending. Budgets help a firm plan, control money, and check performance. Two types are:

- **incremental budget** – last year's budget changed by a small amount.
- **zero-based budget** – every cost must be explained from zero each year.

A **variance** is the difference between the budgeted figure and the actual figure:

$$\text{variance} = \text{actual figure} - \text{budgeted figure}$$



favourable: helps profit
(spend less, or earn more than planned)

adverse: hurts profit
(spend more - as in this wages example)

a **budget** is a financial plan; the **variance** shows how far the actual result missed it

A variance is actual minus budget; favourable helps profit, adverse hurts it

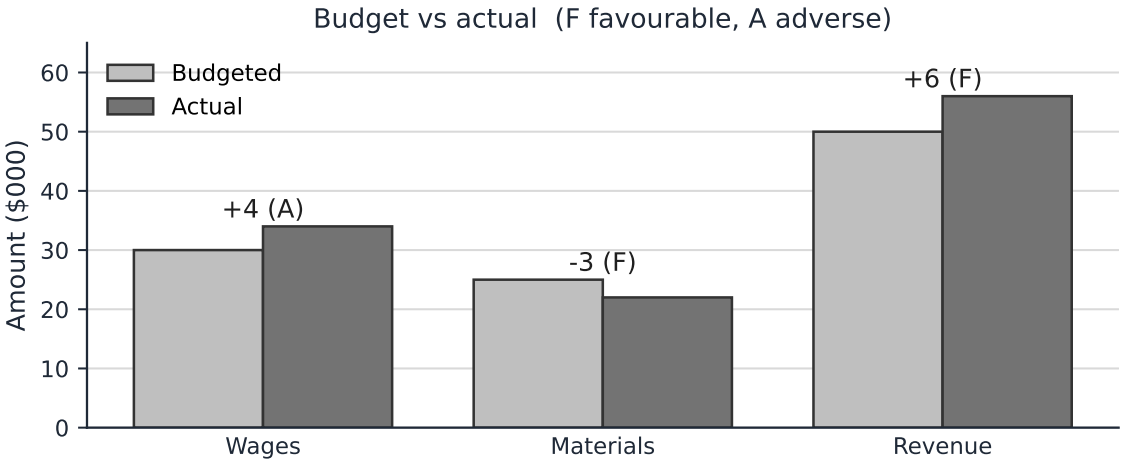
> **Worked example.** A firm budgeted £30,000 for wages but actually spent £34,000. Find the variance and say whether it is favourable or adverse.

$$\text{variance} = £34\,000 - £30\,000 = £4\,000$$

Costs came in higher than planned, so this is an adverse variance.

- a **favourable variance** is better for profit (costs lower, or revenue higher, than planned).
- an **adverse variance** is worse for profit (costs higher, or revenue lower, than planned).

Studying variances shows managers where the plan went wrong, so they can act.



Exam: variance = actual − budget · cost under-spend = F; revenue under-earn = A

Variance = actual − budget: cost overspend is adverse; revenue over-earn is favourable

Exam tips

- Show that **cash is not profit** – a profitable firm can still fail from a cash-flow shortage.
- For **break-even**: contribution per unit = price − variable cost; break-even = fixed costs ÷ contribution; margin of safety = actual − break-even output.
- Match a **source of finance** to the need (short-term for cash flow, long-term or equity for assets) and to the type of business.
- Calculate a **variance** and label it **favourable or adverse** (actual against budget).

Vocabulary card meanings

premises: The buildings and land a firm uses. fixed assets: Long-lasting items a firm owns, such as buildings and machines. internal sources: Finance that comes from inside the firm, such as retained profit. external sources: Finance that comes from outside the firm, such as a bank or new shareholders. short-term: Covering a brief period, often under a year. long-term: Covering several years rather than a few months. sale of assets: Raising money by selling something the firm owns. share capital: Money raised by selling shares in the company. loan: Money borrowed that must be repaid, usually with interest. debenture: A long-term loan certificate issued by a company. leasing: Paying to use an asset that another firm still owns. hire purchase: Paying for an asset in instalments and owning it only after the last payment. venture capital: Money from investors who take a share in a risky young firm. crowdfunding: Raising small sums from many people, usually online. collateral: An asset a borrower offers so

the lender can sell it if the loan is not repaid. closing balance: The amount of cash left at the end of a period. overtrading: Growing sales faster than the firm can finance, so it runs short of cash. indirect costs: Costs that are not tied to one unit of output, such as rent. favourable variance: A difference from the budget that improves profit. adverse variance: A difference from the budget that reduces profit.

Pure Mathematics 1

A-Level Mathematics

This handout covers Topic 1: **Pure Mathematics** 1. It is the algebra and calculus core of the course. Each ## section is one syllabus subtopic.

Quadratics



Under an idealised uniform deck load, a suspension-bridge cable is approximately parabolic.

A **quadratic** is an expression of the form $ax^2 + bx + c$, where $a \neq 0$. The letters a, b, c are the **coefficients** (the fixed numbers). Much of this section is about solving the **equation** $ax^2 + bx + c = 0$.

■ Completing the square

To **complete the square** means to write the quadratic in the form

$$a(x + p)^2 + q.$$

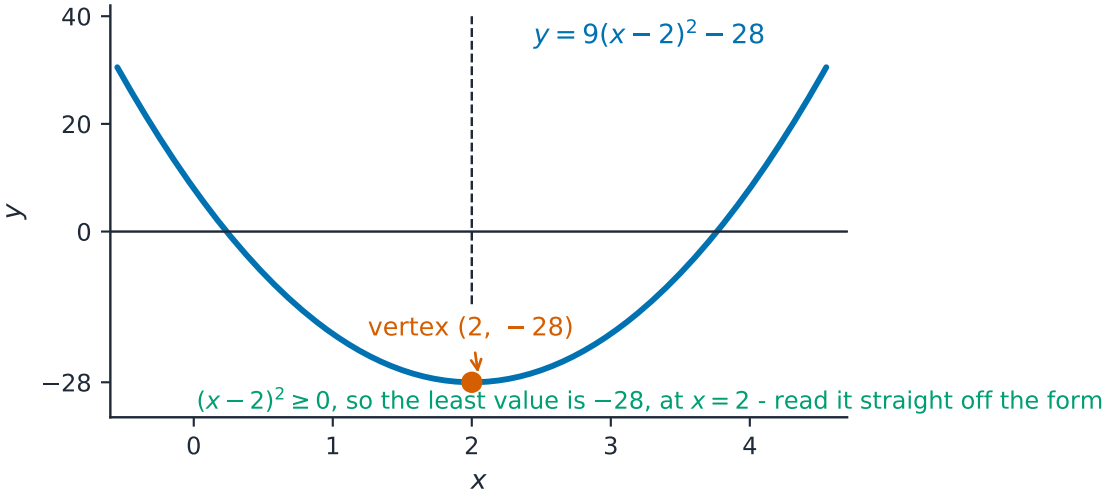
This form is useful: it shows the **vertex** (turning point) of the curve at $(-p,\ q)$, and it gives a quick way to solve the equation.

> **Worked example.** Write $9x^2 - 36x + 8$ in the form $p(x + q)^2 + r$.

Take the factor 9 out of the first two terms , then complete the square inside:

$$\begin{aligned} 9x^2 - 36x + 8 &= 9\left(x^2 - 4x\right) + 8 \\ &= 9\left((x-2)^2 - 4\right) + 8 \\ &= 9(x-2)^2 - 36 + 8 = 9(x-2)^2 - 28. \end{aligned}$$

So $p = 9$, $q = -2$, $r = -28$.



$y = a(x-h)^2 + k$ has vertex (h, k) · min if $a > 0$, max if $a < 0$

The completed square hands you the vertex: least value -28 at x equals 2

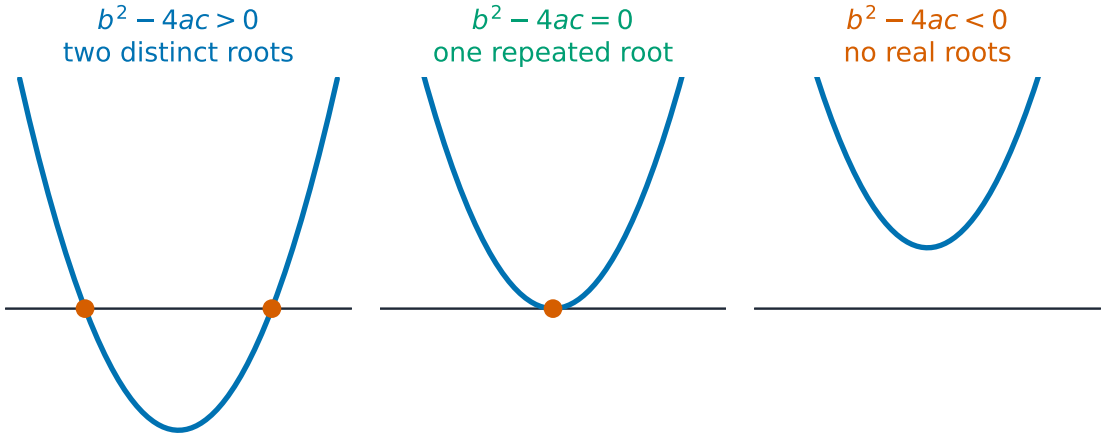
The discriminant

The **discriminant** of $ax^2 + bx + c$ is

$$\Delta = b^2 - 4ac.$$

It tells you how many **real roots** (real solutions) the equation $ax^2 + bx + c = 0$ has:

Discriminant	Roots
$b^2 - 4ac > 0$	two **distinct** real roots
$b^2 - 4ac = 0$	one repeated real root
$b^2 - 4ac < 0$	no real roots



$ax^2 + bx + c = 0 \Rightarrow x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ (real roots need $b^2 - 4ac \geq 0$)

The sign of $b^2 - 4ac$ decides how many times the parabola meets the x-axis.

Worked example. Find the values of the constant k for which $3kx^2 + (k+8)x + 3 = 0$ has two distinct real roots.

Here $a = 3k$, $b = k + 8$, $c = 3$. For two distinct real roots you need $b^2 - 4ac > 0$:

$$(k+8)^2 - 4(3k)(3) > 0 \Rightarrow k^2 + 16k + 64 - 36k > 0 \Rightarrow k^2 - 20k + 64 > 0.$$

Factorise: $(k-4)(k-16) > 0$, so $k < 4$ or $k > 16$. You also need $a \neq 0$, so $k \neq 0$.

Quadratic equations and inequalities

To solve a **quadratic equation**, factorise, complete the square, or use the formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

To solve a **quadratic inequality** such as $(k-4)(k-16) > 0$, find the two roots, then decide which side of each root makes the statement true. A sketch of the parabola helps: an upward-opening curve with two distinct roots is above the x-axis (positive) outside the roots and below it (negative) between them.

Worked inequality. Solve $-x^2 + 4x - 3 \geq 0$. Multiplying by -1 reverses the inequality:

$$x^2 - 4x + 3 \leq 0 \quad \Rightarrow \quad (x-1)(x-3) \leq 0.$$

The factors have opposite signs between the roots, so $1 \leq x \leq 3$. Equality is allowed at both endpoints. Testing $x = 2$ in the original expression gives a positive result; a point outside, such as $x = 0$, gives a negative result. Do not automatically choose the outside intervals when the original leading coefficient is negative.

For a negative leading coefficient, the completed-square constant is a maximum: $-2x^2 + 12x - 11 = -2(x-3)^2 + 7$ has maximum 7 at $x = 3$. Completing the square still works; the sign changes the interpretation.

■ Simultaneous equations

To solve a pair of **simultaneous equations** where one is linear and one is quadratic, use **substitution** : rearrange the linear equation for one letter, then put that into the quadratic. This gives a single quadratic to solve.

Worked substitution. Solve $x + 2y = 7$ and $x^2 + 4y^2 = 25$. The linear equation gives $x = 7 - 2y$, so

$$(7-2y)^2 + 4y^2 = 25 \quad \Rightarrow \quad 2y^2 - 7y + 6 = 0.$$

Factorising gives $(2y-3)(y-2) = 0$, hence $y = \frac{3}{2}$ or 2. Recover the matching values of x: the solutions are $(4, \frac{3}{2})$ and $(3, 2)$. Each pair gives $x + 2y = 7$ and $x^2 + 4y^2 = 25$. Keep the pairing when presenting answers.

When finding intersections, form the common equation first. For $y = x^2 - 2x + 3$ and $y = x + k$, the intersection equation is $x^2 - 3x + 3 - k = 0$ and its discriminant is $4k - 3$. It has no real intersection for $k < \frac{3}{4}$, a repeated intersection for $k = \frac{3}{4}$, and two for $k > \frac{3}{4}$. The boundary point is $(\frac{3}{2}, \frac{9}{4})$.

■ Equations that are quadratic in disguise

Some equations are quadratic in **some function** **of** x. For example $x^4 - 5x^2 + 4 = 0$ is quadratic in x^2: let $u = x^2$, solve $u^2 - 5u + 4 = 0$, then go back to x. You will use this idea again in trigonometry.

Here $(u-1)(u-4) = 0$, so $x^2 = 1$ or 4 and $x = \pm 1, \pm 2$. Always return from the substitute to the original variable.

Worked square-root substitution. For $6x + \sqrt{x} - 1 = 0$, let $v = \sqrt{x} \geq 0$. Then

$$6v^2 + v - 1 = (3v-1)(2v+1) = 0.$$

Only $v = \frac{1}{3}$ is allowed; $v = -\frac{1}{2}$ cannot be a square root. Thus $x = v^2 = \frac{1}{9}$, which satisfies the original equation. Squaring a rejected negative value is not a valid way to recover another solution.

■ Using given roots to find constants

A root is an input giving zero. Substitute each stated root into the original polynomial to obtain simultaneous conditions, or expand its root form and compare coefficients. If the roots are α and β, then

$$a(x-\alpha)(x-\beta) = ax^2 - a(\alpha+\beta)x + a\alpha\beta.$$

Worked example. Suppose $px^2 + 6x - 8$ has roots $2m$ and $-4m$, with $p, m \neq 0$. Expanding $p(x-2m)(x+4m)$ gives $px^2+2pmx-8pm^2$. Hence $pm = 3$ and $pm^2 = 1$, giving $m = \frac{1}{3}$ and $p = 9$. Both actual roots $\frac{2}{3}$ and $-\frac{4}{3}$ satisfy $9x^2 + 6x - 8 = 0$. Do not assume a parameter is positive unless the question states that condition.

Functions

A **function** is a rule that sends each input to exactly one output. Write it as $f(x)$. The **composition of functions** $fg(x)$ means apply g first, then apply f to the result.

- The **domain** is the set of allowed inputs x.
- The **range** is the set of outputs the function actually produces.

A function is **one-one** if different inputs always give different outputs. (No output is repeated.) Only a one-one function has an **inverse function** f^{-1}, which reverses the rule.

The **composition** of two functions means doing one after the other. $fg(x)$ means "do g first, then f": $fg(x) = f(g(x))$. The composite function fg exists only when the range of g lies inside the domain of f.

This condition applies to the whole stated domain of the inner function. If $f(x) = \sqrt{x}$ for $x \geq 0$ and $g(x) = x - 3$ for all real x, $fg(x) = \sqrt{x-3}$ is not defined on all of that inner domain. A new restriction $x \geq 3$ makes the formula valid there; state that change explicitly.

To find a range, check whether a boundary is attained. For $x \geq 1$, the range of $1/x$ is $0 < y \leq 1$; zero is approached but never attained. A restricted quadratic also needs its actual domain: $(x-1)^2 + 2$ for $x \geq 3$ has range $y \geq 6$, since its vertex input is excluded.

An open-endpoint condition. Let $F(x) = 1/(x-2)$ for $x > 2$ and $G(x) = 3x + 2$ for $x > a$. The range of G is $(3a + 2, \infty)$. It lies wholly in the domain of F when $3a + 2 \geq 2$, so $a \geq 0$. Equality is allowed because the lower endpoint of that range is not attained. For $a < 0$, the allowed input 0 gives the forbidden output $G(0) = 2$.

■ Finding an inverse

To find f^{-1}: write $y = f(x)$, make x the subject, then swap letters.

Worked example. The function $f(x) = (x+3)^2 - 12$ is defined for $x \geqslant 0$. Find $f^{-1}(x)$.

Write $y = (x+3)^2 - 12$ and solve for x:

$$(x+3)^2 = y + 12 \;\Rightarrow\; x + 3 = \sqrt{y+12} \;\Rightarrow\; x = \sqrt{y+12} - 3.$$

You take the **positive** square root because $x \geqslant 0$ means $x + 3 \geqslant 3 > 0$. So

$$f^{-1}(x) = \sqrt{x+12} - 3.$$

The original range is $[-3, \infty)$, attained from original inputs $x \geq 0$. Therefore the inverse has domain $x \geq -3$ and range $f^{-1}(x) \geq 0$. Merely requiring the radical to be real would give the larger, incorrect inverse domain $x \geq -12$.

A different restriction can require the negative square-root branch. For $h(x) = (x + 1)^2 - 5$ with $x \leq -1$, solve $y + 5 = (x + 1)^2$ using $x + 1 \leq 0$:

$$h^{-1}(x) = -1 - \sqrt{x + 5}, \qquad x \geq -5.$$

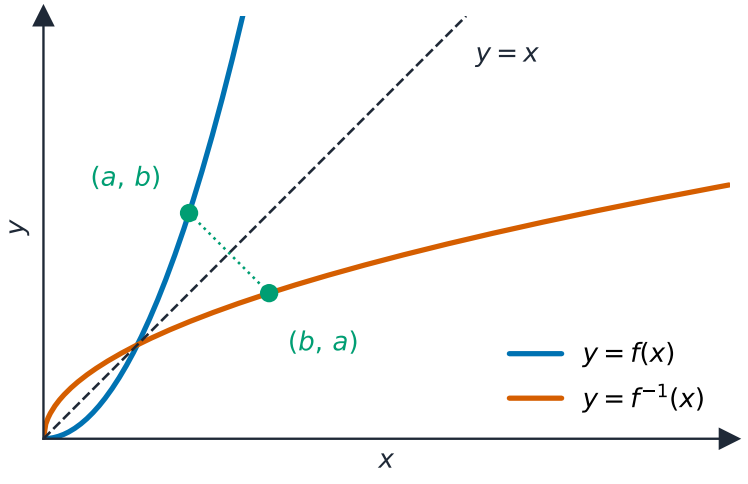
Its range is $h^{-1}(x) \leq -1$. The endpoint $(-1, -5)$ reflects to $(-5, -1)$ and the point $(-3, -1)$ reflects to $(-1, -3)$. A $\pm$ answer would not define a one-one inverse function. The inverse is also different from the reciprocal $1/h(x)$.

To invert a composite of compatible one-one functions, reverse their order: $(gf)^{-1} = f^{-1}g^{-1}$. Alternatively form the composite rule first, then invert it with the original domain restriction.

For example, $f(x) = (x - 1)^2 + 2$ for $x \leq 1$ and $g(x) = 5 - x$ give $H(x) = gf(x) = 3 - (x - 1)^2$, with range $H(x) \leq 3$. Solving $y = 3 - (x - 1)^2$ using $x - 1 \leq 0$ gives $H^{-1}(x) = 1 - \sqrt{3 - x}$ for $x \leq 3$, with range $H^{-1}(x) \leq 1$. Composing and then inverting keeps the whole branch condition visible.

■ Graphs of inverses and transformations

The graph of $y = f^{-1}(x)$ is the **reflection** of $y = f(x)$ in the line $y = x$.



inverse = reflection in $y = x$ · domain of f^{-1} = range of f

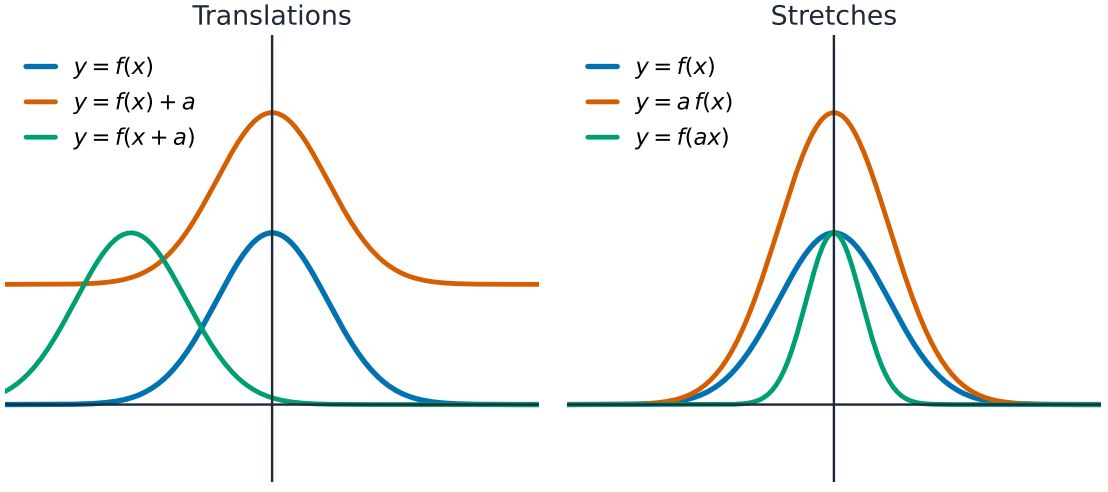
Reflecting $y = f(x)$ in the line $y = x$ gives its inverse; a point (a, b) becomes (b, a).

You should know these **transformations** of $y = f(x)$: In the stretch descriptions below take $a > 0$. Negative multipliers also require a reflection : $-f(x)$ reflects in the x-axis and $f(-x)$ reflects in the y-axis.

New equation	Effect on the graph
$y = f(x) + a$	**translation** up by a
$y = f(x + a)$	translation left by a
$y = a\,f(x)$	**stretch** in the y-direction, scale factor a
$y = f(ax)$	stretch in the x-direction, scale factor $\frac{1}{a}$

When two transformations are combined, the order can matter. State each one fully (type, direction, and amount).

For $g(x) = 2f(x+3) - 1$, use two transformations: stretch vertically by factor 2, then translate by $(-3, -1)$ (left 3 and down 1 together). A point $(4, 5)$ becomes $(4, 10)$, then $(1, 9)$. Reversing these same operations gives $2(f(x + 3) - 1) = 2f(x + 3) - 2$, a different graph. If two transformations are required, combine the horizontal and vertical shifts into one translation. For a horizontal multiplier $f(ax)$, a point (s, t) moves to $(s/a, t)$; the scale factor is reciprocal, not a.



$f(x) + a$ moves up · $f(x + a)$ moves left · $af(x)$ stretch y · $f(ax)$ compress x if $a > 1$

Adding to the output or input slides the curve; a multiplier stretches it.

Coordinate geometry

Coordinate geometry connects positions, distances and intersections with algebra. A straight path can cross a circular boundary twice, touch it once or miss it. Solve their equations together to distinguish these cases exactly.

■ Choose and check a line equation

For two points with different x-coordinates, the **gradient** is

$$m = \frac{y_2 - y_1}{x_2 - x_1}.$$

Useful forms of an **equation of a straight line** are

$$y = mx + c, \qquad y - y_1 = m(x - x_1), \qquad ax + by + c = 0.$$

For example, through $(2, 3)$ and $(6, 11)$ the gradient is 2. The same line is $y - 3 = 2(x - 2)$, $y = 2x - 1$, or $2x - y - 1 = 0$. Substitute each given point to check. In $2x - 3y + 6 = 0$, rearranging gives $y = \frac{2}{3}x + 2$, so the gradient is $2/3$, not 2.

Distinct non-vertical **parallel** lines have equal gradients. For **perpendicular** lines with finite gradients, $m_1 m_2 = -1$. Horizontal lines $y = h$ have gradient zero; vertical lines $x = k$ have undefined gradient. They are perpendicular, but the product rule cannot be used for a vertical line. Through $(4, -2)$, the horizontal and vertical lines are $y = -2$ and $x = 4$.

■ Distances, midpoints and perpendicular bisectors

Pythagoras gives the distance between two points:

$$PQ = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

The **midpoint** is

$$M = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right).$$

A **perpendicular bisector** passes through this midpoint at right angles to the segment . Every point on it is equally far from the two ends.

> **Worked example.** For $P(-1, 2)$ and $Q(5, 6)$, $PQ = \sqrt{6^2 + 4^2} = 2\sqrt{13}$ and $M = (2, 4)$. The segment gradient is $2/3$, so the bisector gradient is $-3/2$. Its equation is
>
> $$y - 4 = -\frac{3}{2}(x - 2) \quad \Longrightarrow \quad 3x + 2y = 14.$$

The midpoint alone is not an equation, and a perpendicular through an endpoint is not a bisector. If PQ is horizontal, its bisector is the vertical line through M; do not take a reciprocal of zero.

■ Read a circle in two forms

A **circle** with **centre** (a, b) and positive **radius** r has equation

$$(x-a)^2 + (y-b)^2 = r^2.$$

An expanded equation needs both squares completed:

$$\begin{aligned} x^2 + y^2 - 6x + 10y - 27 &= 0, \\ (x-3)^2 - 9 + (y+5)^2 - 25 &= 27, \\ (x-3)^2 + (y+5)^2 &= 61. \end{aligned}$$

The centre is $(3, -5)$ and radius $\sqrt{61}$. More generally, $x^2 + y^2 + 2gx + 2fy + c = 0$ gives centre $(-g, -f)$ and $r^2 = g^2 + f^2 - c$. Check that $r^2 > 0$ for a real circle of positive radius. If it is negative there are no real points; if zero there is only one point. Substitute a proposed point into the equation to test it.

■ Use a diameter , a semicircle or symmetry

> **Worked example.** The points $P(1, 1)$ and $Q(7, 11)$ are the ends of a **diameter**. Its midpoint is $(4, 6)$ and its half-length is
>
> $$r = \frac{1}{2}\sqrt{6^2 + 10^2} = \sqrt{34}.$$

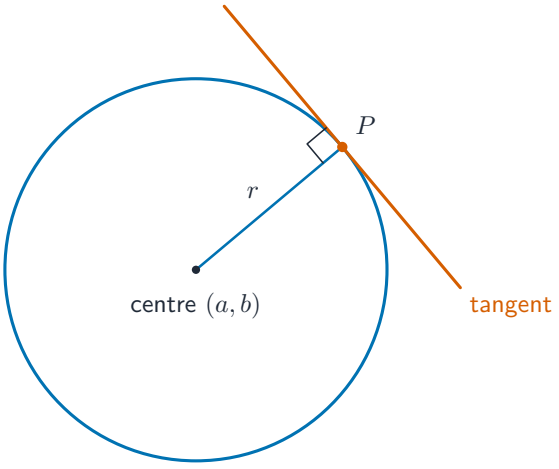
The circle is $(x-4)^2 + (y-6)^2 = 34$. A general chord's midpoint is not necessarily the centre: the diameter condition matters.

The angle formed at any other point on this circle by lines to P and Q is $90°$: it is an angle in a semicircle. For a non-diameter chord, the line from the centre to its midpoint is perpendicular to the chord and is a symmetry line. Use this to recover a reflected intersection point or a missing coordinate.

For instance, $(x-2)^2+(y-1)^2 = 5$ has diameter ends $A(0, 0)$ and $B(4, 2)$. Point $D(4, 0)$ lies on it and $\angle ADB = 90°$. The chord AD has midpoint $(2, 0)$, so its perpendicular bisector $x = 2$ passes through the centre and reflects A onto D.

■ Tangents : known and unknown contact points

A **tangent** is perpendicular to the radius at the contact point.



A tangent meets the radius at a right angle.

For the circle centred at $(2, 1)$ above, the radius to $B(4, 2)$ has gradient $1/2$. The tangent has gradient -2, so $y-2=-2(x-4)$, or $2x+y=10$. Use B in this equation, not the centre. At the top and bottom the tangents are horizontal, $y=1\pm\sqrt{5}$; at the left and right they are vertical, $x=2\pm\sqrt{5}$.

> **Worked example: unknown contact points.** Find both points on this circle where the tangent gradient is -2. The radius gradient must be $1/2$, and the radius line through the centre is $y-1=\frac{1}{2}(x-2)$. Put $u=x-2$ and substitute:

$$u^2+\frac{u^2}{4}=5 \quad\Longrightarrow\quad u=\pm 2.$$

The contact points are $(4, 2)$ and $(0, 0)$, with tangent equations $2x+y=10$ and $2x+y=0$. A given tangent direction normally gives two opposite contact points. Keep exact square roots when the solutions are not integers. Radius geometry and substitution suffice; circle differentiation is not required here.

■ Intersections and parameter boundaries

Substitute a line into a circle to get a quadratic, solve for one coordinate and recover the other. In $(x-2)^2+(y-1)^2=5$, the line $y=0$ gives $(x-2)^2=4$, hence the points $(0, 0)$ and $(4, 0)$.

For the family $y=k$, the same circle gives

$$(x-2)^2=5-(k-1)^2.$$

There are two points when $1-\sqrt{5}<k<1+\sqrt{5}$, one tangent point at each boundary, and no points outside that interval. At either boundary $x=2$. For a general sloping line, the resulting quadratic's discriminant is positive, zero or negative in the same three cases. A repeated root is one point.

For a line and a quadratic curve, equate the two expressions for y. For $y = x^2$ and $y = 2x + k$:

$$x^2 - 2x - k = 0, \qquad \Delta = 4 + 4k.$$

Thus there are two intersections for $k > -1$, one for $k = -1$ at $(1, 1)$, and none for $k < -1$. Show the algebraic condition; a sketch alone does not prove an exact boundary. Some families have a positive discriminant for every real parameter, so they always intersect twice. Check separately if a parameter could make the quadratic coefficient zero.

■ Build a circle from axis gaps

A circle entirely in the first quadrant has centre (a, b) and radius r. Its least distances from the y-axis and x-axis are $a - r$ and $b - r$, respectively.

> **Worked example.** The gap from the y-axis is 1, the gap from the x-axis is 3, and the distance from the origin to the centre is $\sqrt{34}$. Then the centre is $(r + 1, r + 3)$, so
>
> $$\begin{aligned}(r + 1)^2 + (r + 3)^2 &= 34,\\ r^2 + 4r - 12 &= 0,\\ (r + 6)(r - 2) &= 0.\end{aligned}$$
>
> Reject the negative radius. Therefore $r = 2$, centre $(3, 5)$, and equation $(x-3)^2+(y-5)^2 = 4$. The point furthest from the origin lies along the ray from the origin through the centre: its radius has gradient $5/3$, so its tangent gradient is $-3/5$.

Independent check. A different circle's least gaps from the x- and y-axes are 6 and 2, and its centre is $4\sqrt{5}$ from the origin. Find the radius, centre, circle equation and tangent gradient at its furthest point from the origin. Set up the centre coordinates before calculating.

Check after attempting: centre $(r + 2, r + 6)$ gives $(r + 2)^2 + (r + 6)^2 = 80$. Thus $(r + 10)(r - 2) = 0$, $r = 2$, centre $(4, 8)$ and equation $(x - 4)^2 + (y - 8)^2 = 4$. The radius direction has gradient 2, so the tangent gradient is $-1/2$.

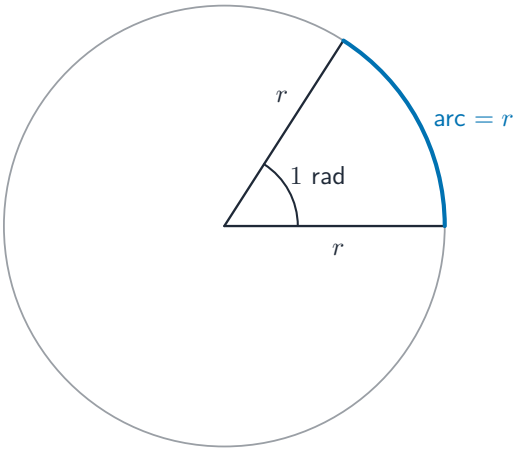
Circular measure

■ Radians

A **radian** is another way to measure angles. One radian is the angle at the centre of a circle that cuts off an **arc** equal in length to the radius. The link between radians and **degrees** is

$$\pi \text{ radians} = 180^\circ.$$

So to change degrees to radians, multiply by $\dfrac{\pi}{180}$; to change radians to degrees, multiply by $\dfrac{180}{\pi}$.



one **radian**: the angle whose arc is exactly one radius long

π rad $= 180°$, so 1 rad $\approx 57.3°$

degrees $\to$ radians: $\times \frac{\pi}{180}$ radians $\to$ degrees: $\times \frac{180}{\pi}$

One radian: the angle whose arc equals the radius

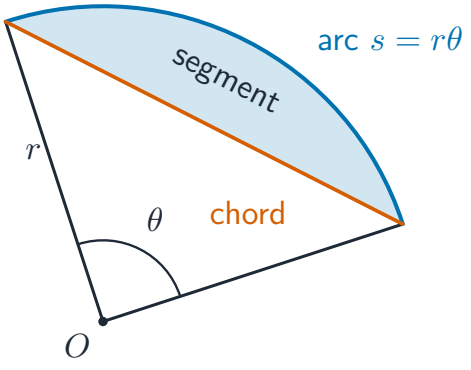
■ Arc length and sector area

For a **sector** with radius r and angle θ **in radians**:

$$\text{arc length} = s = r\theta, \qquad \text{sector area} = A = \tfrac{1}{2}r^2\theta.$$

A **chord** cuts the sector into a triangle and a **segment**. The segment area is the sector minus the triangle:

$$\text{segment} = \tfrac{1}{2}r^2\theta - \tfrac{1}{2}r^2\sin\theta = \tfrac{1}{2}r^2(\theta - \sin\theta).$$



The shaded segment is the part of the sector between the chord and the arc.

Worked example. A sector has centre O and the angle at O is $\frac{2}{3}\pi$ radians. Show that the segment cut off by the chord has area about $0.614r^2$.

$$\text{segment} = \tfrac{1}{2}r^2\left(\tfrac{2}{3}\pi - \sin\tfrac{2}{3}\pi\right) = \tfrac{1}{2}r^2(2.0944 - 0.8660) = \tfrac{1}{2}r^2(1.2284) \approx 0.614r^2.$$

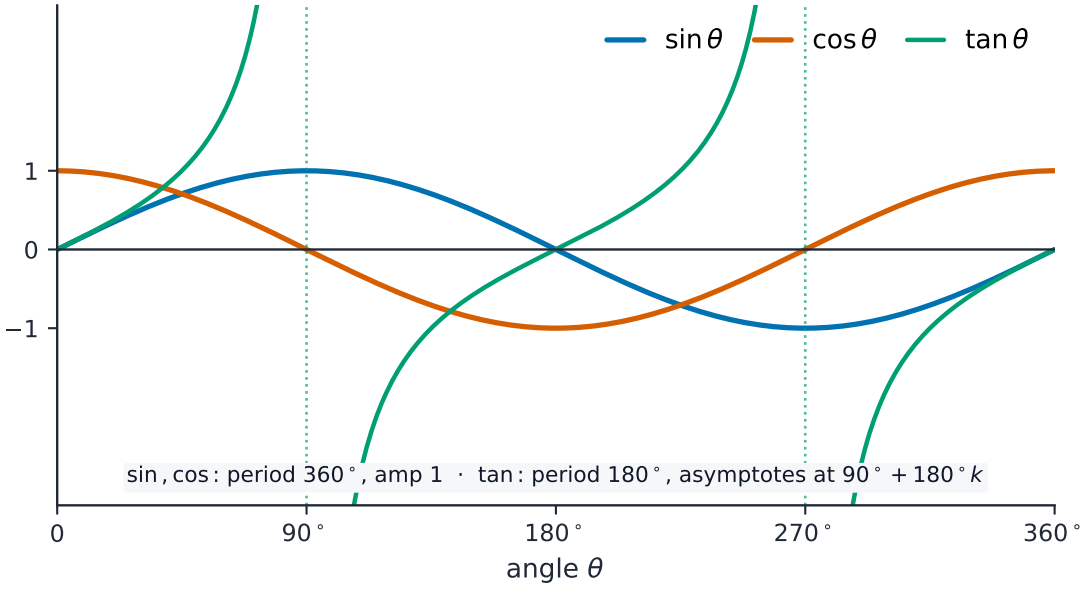


A Ferris wheel: a point on the rim rises and falls like a sine curve.

Graphs and exact values

You must know the shape of the graphs of the **sine**, **cosine** and **tangent function** (written sin, cos, tan). The sine and cosine graphs wave between -1 and 1 and repeat every $360°$ (2π). Learn these exact values:

θ	$30°$	$45°$	$60°$
$\sin\theta$	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{\sqrt{3}}{2}$
$\cos\theta$	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$
$\tan\theta$	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$



Over one turn sin *and* cos *stay between* -1 *and* 1*;* tan *shoots off at* $90°$ *and* $270°$*.*

The notations $\sin^{-1} x$, $\cos^{-1} x$, $\tan^{-1} x$ mean the **inverse** angle (the **principal value**).

■ Identities

An **identity** is true for every value of the angle. The two you must know are

$$\tan\theta \equiv \frac{\sin\theta}{\cos\theta}, \qquad \sin^2\theta + \cos^2\theta \equiv 1.$$

Use them to rewrite an equation so it contains only one trig function.

■ Solving trigonometric equations

To solve a **trigonometric equation**, first reduce it to one function, then find every solution in the given interval.

Worked example. Solve $6\sin\theta = 1 + \dfrac{2}{\sin\theta}$ for $-180° < \theta < 180°$.

Multiply through by $\sin\theta$ to clear the fraction. This makes a **quadratic in** $\sin\theta$:

$$6\sin^2\theta - \sin\theta - 2 = 0 \;\Rightarrow\; (3\sin\theta - 2)(2\sin\theta + 1) = 0.$$

So $\sin\theta = \frac{2}{3}$ or $\sin\theta = -\frac{1}{2}$.

- $\sin\theta = \frac{2}{3}$: $\theta = 41.8°$ or $\theta = 180° - 41.8° = 138.2°$.
- $\sin\theta = -\frac{1}{2}$: $\theta = -30°$ or $\theta = -150°$.

The four solutions are $\theta = -150°,\ -30°,\ 41.8°,\ 138.2°$.

The binomial expansion

For a positive integer n, the **binomial expansion** is

$$(a+b)^n = a^n + \binom{n}{1}a^{n-1}b + \binom{n}{2}a^{n-2}b^2 + \cdots + b^n,$$

where $\binom{n}{r} = \dfrac{n!}{r!\,(n-r)!}$ is a **binomial coefficient** .

Worked example. Find the first three terms, in ascending powers of x, of $(2-px)^5$.

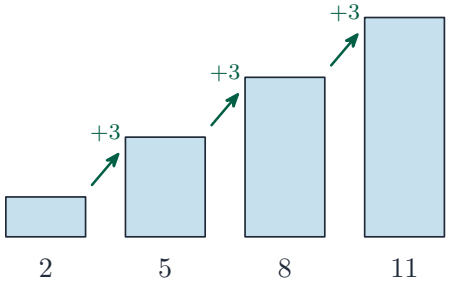
$$(2-px)^5 = 2^5 + \binom{5}{1}2^4(-px) + \binom{5}{2}2^3(-px)^2 + \cdots = 32 - 80px + 80p^2x^2 + \cdots$$

Arithmetic and geometric progressions

A **progression** (sequence) is a list of terms following a rule.

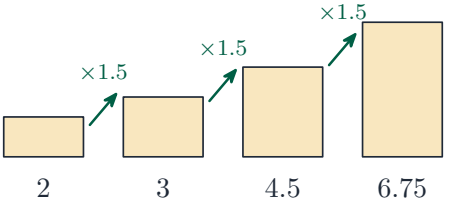
- An **arithmetic progression** (AP) adds a fixed **common difference** d each step. The nth **term** is $u_n = a + (n-1)d$, and the sum of the first n terms is $S_n = \frac{n}{2}\big(2a + (n-1)d\big)$.
- A **geometric progression** (GP) multiplies by a fixed **common ratio** r each step. The nth term is $u_n = ar^{n-1}$, and $S_n = \dfrac{a(1-r^n)}{1-r}$.

arithmetic: add d each step



$u_n = a + (n-1)d$: equal steps

geometric: multiply by r each step



$u_n = ar^{n-1}$: growing steps

An AP climbs in equal steps; a GP's steps grow by the same ratio

A GP is **convergent** – it **converges** (settles to a limit) – when $|r| < 1$. Then it has a **sum to infinity**

$$S_\infty = \frac{a}{1-r}.$$

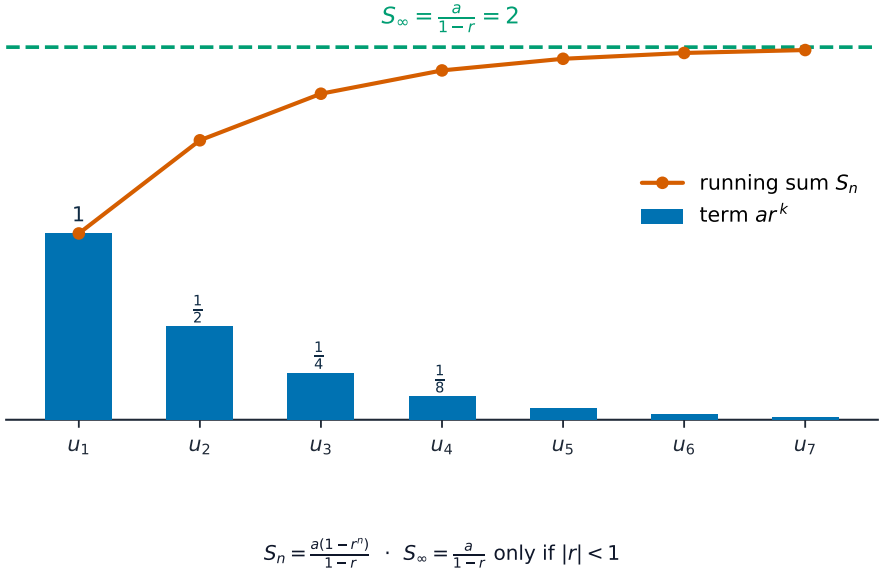
Worked example. The third term of a GP is 18 and the sum of the first three terms is 26. The common ratio is negative. Find the sum to infinity.

From $ar^2 = 18$ you get $a = \dfrac{18}{r^2}$. Put this into $a(1 + r + r^2) = 26$:

$$18(1 + r + r^2) = 26r^2 \;\Rightarrow\; 8r^2 - 18r - 18 = 0 \;\Rightarrow\; (4r + 3)(r - 3) = 0.$$

The ratio is negative, so $r = -\frac{3}{4}$ and $a = \dfrac{18}{(3/4)^2} = 32$. Then

$$S_\infty = \frac{32}{1 - (-\frac{3}{4})} = \frac{32}{\frac{7}{4}} = \frac{128}{7}.$$



Convergent GP: bars shrink toward zero while the running sum approaches S infinity = a/(1-r)

Differentiation

Differentiation finds the **gradient of a curve** at each point. The gradient is the **limit** of the gradients of shorter and shorter **chords**, called the **derivative** .

The rules

Write the derivative as $f'(x)$ or $\dfrac{dy}{dx}$. The basic rule is

$$\frac{d}{dx}(x^n) = nx^{n-1} \quad \text{for any rational } n.$$

Differentiate sums term by term, and use the **chain rule** for a function inside a function:

$$\frac{d}{dx} f(g(x)) = f'(g(x)) \cdot g'(x).$$

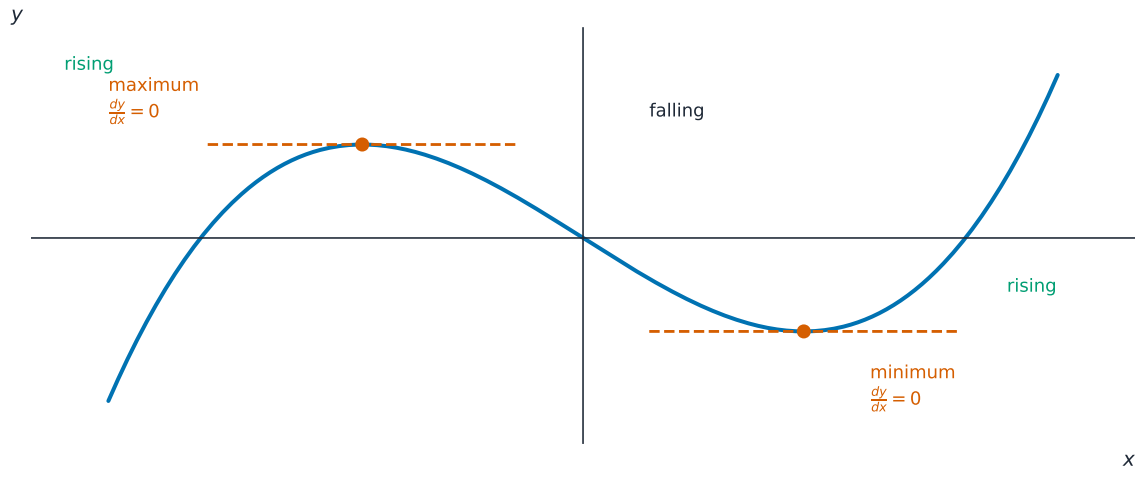
Differentiating again gives the **second derivative** $f''(x)$ or $\dfrac{d^2y}{dx^2}$.

Using the derivative

- **Tangent and normal .** The gradient of the curve at a point is the gradient of the **tangent** there. The **normal** is perpendicular to the tangent, so its gradient is $-\dfrac{1}{(\text{tangent gradient})}$.
- **Increasing or decreasing.** The function is an **increasing function** where $\dfrac{dy}{dx} > 0$, and a **decreasing function** where $\dfrac{dy}{dx} < 0$.
- **Rate of change .** A derivative is a **rate of change**. Linked rates use the chain rule, e.g. $\dfrac{dy}{dt} = \dfrac{dy}{dx} \cdot \dfrac{dx}{dt}$.

Stationary points

A **stationary point** is where $\dfrac{dy}{dx} = 0$. Test its nature with the second derivative: $f''(x) > 0$ gives a **minimum point** , and $f''(x) < 0$ gives a **maximum point** . When $f''(x) = 0$ the test is inconclusive: the point may be a **point of inflexion** , where the curve changes concavity (the way it bends) – check the sign of $\dfrac{dy}{dx}$ just before and after to decide.



stationary points: $\frac{dy}{dx} = 0$ · use $\frac{d^2y}{dx^2}$ or first-derivative sign to classify

At a maximum or a minimum the tangent is flat, so $\frac{dy}{dx} = 0$.

Worked example. The curve $y = 4x^{1/2} - x$ has a maximum point at $x = a$. Find a.

$$\frac{dy}{dx} = 2x^{-1/2} - 1 = 0 \;\Rightarrow\; \frac{2}{\sqrt{x}} = 1 \;\Rightarrow\; \sqrt{x} = 2 \;\Rightarrow\; x = 4.$$

So $a = 4$.

Integration

Integration is the reverse of differentiation. Reversing the power rule gives

$$\int (ax + b)^n \, dx = \frac{(ax + b)^{n+1}}{a(n + 1)} + C \quad (n \neq -1).$$

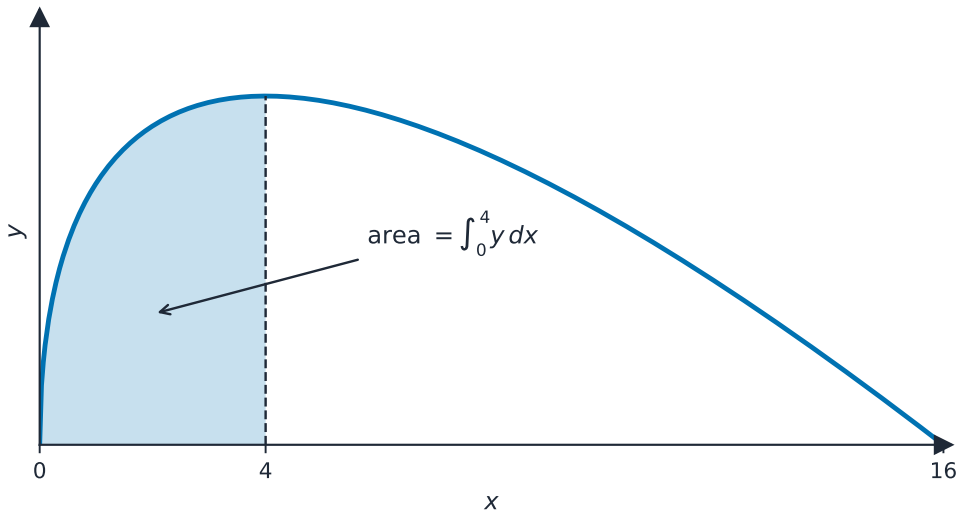
The $+C$ is the **constant of integration** . If you know one point on the curve, substitute it to find C.

■ Definite integrals and area

A **definite integral** has limits and gives a number:

$$\int_p^q f(x) \, dx = \big[F(x)\big]_p^q = F(q) - F(p).$$

The area of the **region** between a curve and the x-axis, from $x = p$ to $x = q$, is $\int_p^q y \, dx$. For the area between two curves, integrate (top curve − bottom curve).



$\int_a^b f(x)\,dx =$ net signed area between $y = f(x)$ and the x-axis

The definite integral $\int_0^4 y\,dx$ is the shaded area under the curve.

Worked example. The curve $y = 4x^{1/2} - x$ meets the x-axis again at $x = 16$. Find the area between the curve and the x-axis from $x = 0$ to $x = 4$.

$$\int_0^4 \left(4x^{1/2} - x\right) dx = \left[\frac{8}{3}x^{3/2} - \frac{x^2}{2}\right]_0^4 = \frac{8}{3}(8) - \frac{16}{2} = \frac{64}{3} - 8 = \frac{40}{3}.$$

An **improper integral** has an infinite limit or an endpoint where the integrand is undefined; evaluate it as a **limit**. For example,

$$\int_1^\infty x^{-2}\,dx = \lim_{b\to\infty}\left[-\frac{1}{x}\right]_1^b = \lim_{b\to\infty}\left(1 - \frac{1}{b}\right) = 1,$$

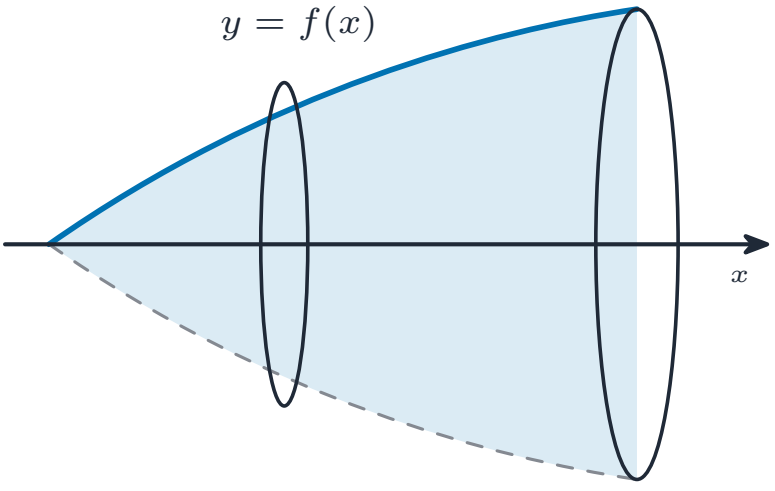
and $\displaystyle\int_0^1 x^{-1/2}\,dx = \left[2x^{1/2}\right]_0^1 = 2$, which is finite even though the integrand blows up at $x = 0$.

Volume of revolution

When a region is turned all the way around an axis it sweeps out a solid. The **volume of revolution** about the x-axis is

$$V = \pi \int_p^q y^2\,dx,$$

and about the y-axis it is $V = \pi \displaystyle\int x^2\,dy$. For example, the region under $y = \sqrt{x}$ from $x = 0$ to $x = 4$, turned about the x-axis, has volume $\pi \displaystyle\int_0^4 x\,dx = \pi\left[\tfrac{x^2}{2}\right]_0^4 = 8\pi$.



rotate the region around the x-axis to sweep a solid

Rotating the region under a curve around the x-axis sweeps out a solid; each thin slice is a disc of area πy^2, so $V = \pi \int y^2 \, dx$

Exam tips

- Show **every line of algebra** – method marks are lost by jumping steps; use the **discriminant** $b^2 - 4ac$ to decide the number of real roots.
- Work in **radians** for circular measure (arc $= r\theta$, sector area $= \frac{1}{2}r^2\theta$) and for calculus of trig functions.
- For **differentiation**, set $\frac{dy}{dx} = 0$ for stationary points and use the second derivative to classify them.
- For **integration**, add $+c$ to an indefinite integral and use limits for area; an area below the axis gives a negative integral.

Pure Mathematics 2

A-Level Mathematics

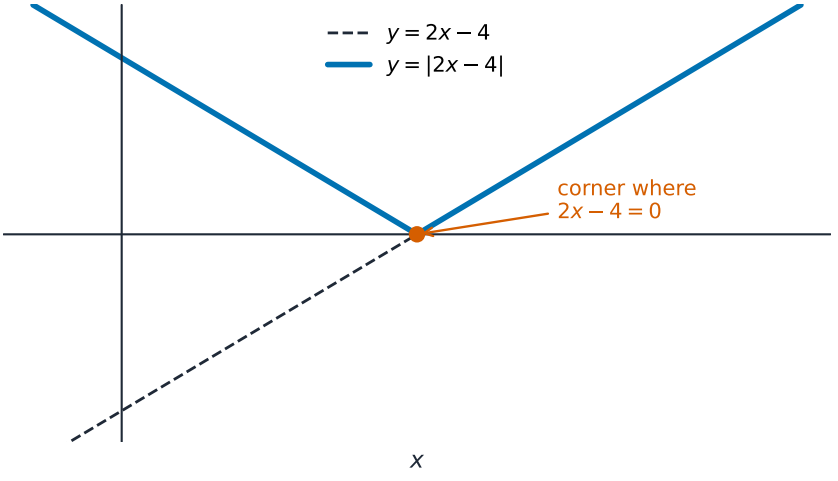
This handout covers Topic 2: **Pure Mathematics** 2. It adds the modulus and polynomial algebra, **logarithms** and the **exponential function** , more trigonometry, and new ways to differentiate and integrate.

Algebra

The modulus

The **modulus** $|x|$ is the size of a number with its sign removed, so $|x| \geqslant 0$ always. The graph of $y = |ax + b|$ is a "V" shape that bounces off the x-axis. Two useful rules for solving equations and inequalities are

$$|a| = |b| \;\Leftrightarrow\; a^2 = b^2, \qquad |x - a| < b \;\Leftrightarrow\; a - b < x < a + b.$$



$y = |f(x)|$ reflects parts of $y = f(x)$ below the x-axis above it

Taking the modulus folds the part of the line below the axis upward into a V.

Worked example. Solve $|3x + 8| < 9$.

Using the second rule with the inequality written as $-9 < 3x + 8 < 9$:

$$-9 < 3x + 8 < 9 \;\Rightarrow\; -17 < 3x < 1 \;\Rightarrow\; -\tfrac{17}{3} < x < \tfrac{1}{3}.$$

■ Polynomial division and the factor and remainder theorems

A **polynomial** is a sum of powers of x, such as $2x^4+3x^2-5$. Its **degree** is the highest power. When you divide one polynomial by another you get a **quotient** and a **remainder** .

- **Remainder theorem** : the remainder when $p(x)$ is divided by $(x-a)$ is $p(a)$.
- **Factor theorem** : $(x-a)$ is a factor of $p(x)$ exactly when $p(a)=0$.

> **Worked example.** The polynomial $p(x)=2x^4+kx^3+kx^2+17x+18$ has factor $(x+2)$. Find k.

By the factor theorem $p(-2)=0$:

$$2(16)+k(-8)+k(4)+17(-2)+18=0 \;\Rightarrow\; 16-4k=0 \;\Rightarrow\; k=4.$$

Logarithmic and exponential functions



Earthquake strength is measured on the logarithmic Richter scale.

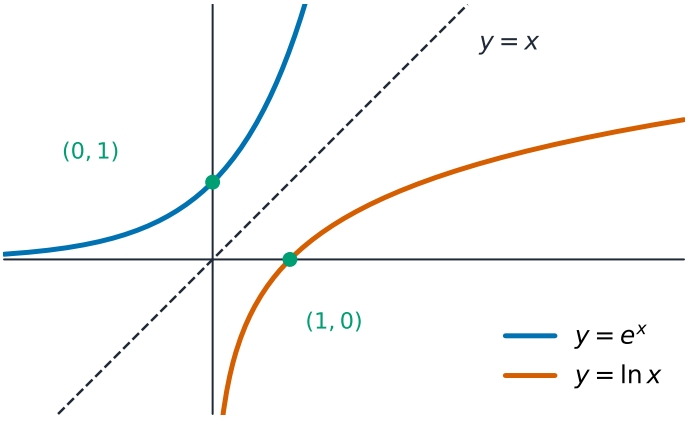


Populations can grow exponentially when resources are plentiful.

A **logarithm** answers the question "what power?". If $a^x = y$ then $x = \log_a y$. Logarithms and **indices** (powers) are reverse ideas. The **laws of logarithms** are

$$\log(mn) = \log m + \log n, \qquad \log\frac{m}{n} = \log m - \log n, \qquad \log(m^k) = k \log m.$$

The exponential function e^x and the **natural logarithm** $\ln x$ are inverse functions, so $\ln(e^x) = x$ and $e^{\ln x} = x$. When the unknown is in the power, take logs of both sides.



ln and exp inverse · reflect in $y = x$

e^x and $\ln x$ undo each other, so each is the other reflected in $y = x$.

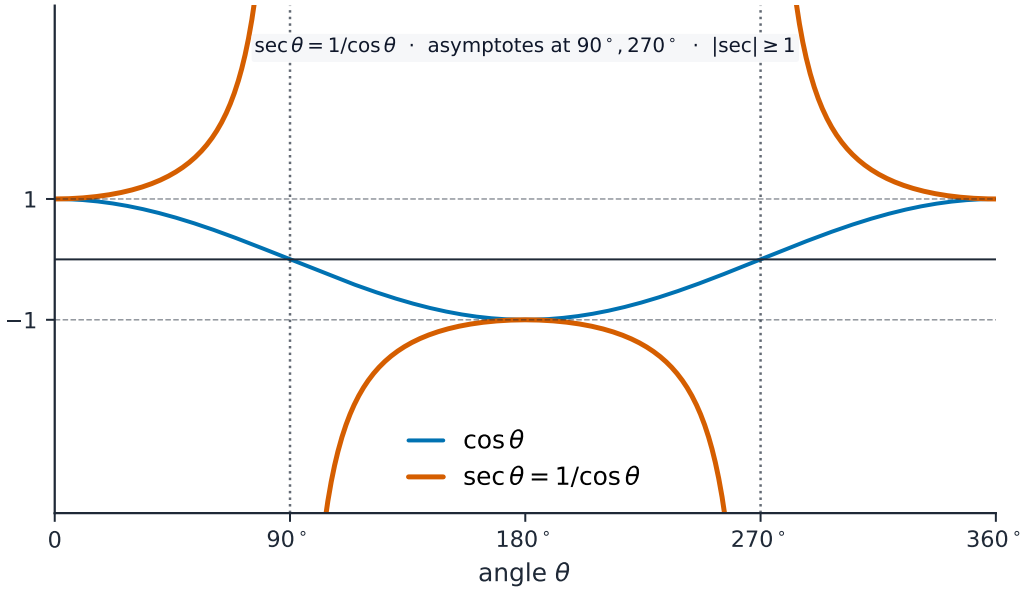
Worked example. Solve $4^x < 0.05$.

$$4^x < 0.05 \;\Rightarrow\; x \ln 4 < \ln 0.05 \;\Rightarrow\; x < \frac{\ln 0.05}{\ln 4} = -2.16 \text{ (3 s.f.)}.$$

Linear form : a relationship like $y = Ax^n$ becomes a straight line if you take logs: $\ln y = \ln A + n \ln x$. Plotting $\ln y$ against $\ln x$ gives a line with gradient n and intercept $\ln A$, so you can find the unknown constants.

Trigonometry

There are three more functions, each the reciprocal of one you know: the **secant** $\sec\theta = \dfrac{1}{\cos\theta}$, the **cosecant** $\csc\theta = \dfrac{1}{\sin\theta}$, and the **cotangent** $\cot\theta = \dfrac{1}{\tan\theta}$.



$\sec\theta = 1/\cos\theta$ *rises to an asymptote wherever* $\cos\theta = 0$.

You must know these **trigonometric identities** and choose the right one for each problem:

$$\sec^2\theta \equiv 1 + \tan^2\theta, \qquad \csc^2\theta \equiv 1 + \cot^2\theta.$$

You also use the **compound angle** formulae for $\sin(A \pm B)$, $\cos(A \pm B)$, $\tan(A \pm B)$, the **double angle** formulae

$$\sin 2A = 2\sin A\cos A, \quad \cos 2A = 2\cos^2 A - 1, \quad \tan 2A = \frac{2\tan A}{1 - \tan^2 A},$$

and the **R-formula** $a\sin\theta + b\cos\theta = R\sin(\theta + \alpha)$, where $R = \sqrt{a^2 + b^2}$ and $\tan\alpha = \dfrac{b}{a}$.

Worked example. Solve $2\tan^2\theta + 3\sec\theta = 18$ for $-180° < \theta < 180°$.

Replace $\tan^2\theta$ with $\sec^2\theta - 1$ to get one function:

$$2(\sec^2\theta - 1) + 3\sec\theta = 18 \;\Rightarrow\; 2\sec^2\theta + 3\sec\theta - 20 = 0 \;\Rightarrow\; (2\sec\theta - 5)(\sec\theta + 4) = 0.$$

So $\sec\theta = \frac{5}{2}$ or $\sec\theta = -4$, giving $\cos\theta = \frac{2}{5}$ or $\cos\theta = -\frac{1}{4}$. The solutions are $\theta = \pm 66.4°$ and $\theta = \pm 104.5°$.

Differentiation

Learn these standard derivatives:

$$\frac{d}{dx}e^x = e^x, \quad \frac{d}{dx}\ln x = \frac{1}{x}, \quad \frac{d}{dx}\sin x = \cos x, \quad \frac{d}{dx}\cos x = -\sin x, \quad \frac{d}{dx}\tan x = \sec^2 x.$$

For a product or a quotient of two functions, use:

- the **product rule** : $(uv)' = u'v + uv'$;
- the **quotient rule** : $\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$.

When a curve is given by **parametric equations** $x = x(t)$, $y = y(t)$, the gradient is $\frac{dy}{dx} = \frac{dy/dt}{dx/dt}$. When y is defined **implicitly** (not made the subject), differentiate every term with respect to x, using the chain rule on the y terms, then solve for $\frac{dy}{dx}$.

Worked example. Given $y = 6x\cos(x^2 + 1)$, find $\frac{dy}{dx}$.

Use the product rule with $u = 6x$ and $v = \cos(x^2 + 1)$ (and the chain rule for v):

$$\frac{dy}{dx} = 6\cos(x^2 + 1) + 6x \cdot \big(-2x\sin(x^2 + 1)\big) = 6\cos(x^2 + 1) - 12x^2\sin(x^2 + 1).$$

Integration

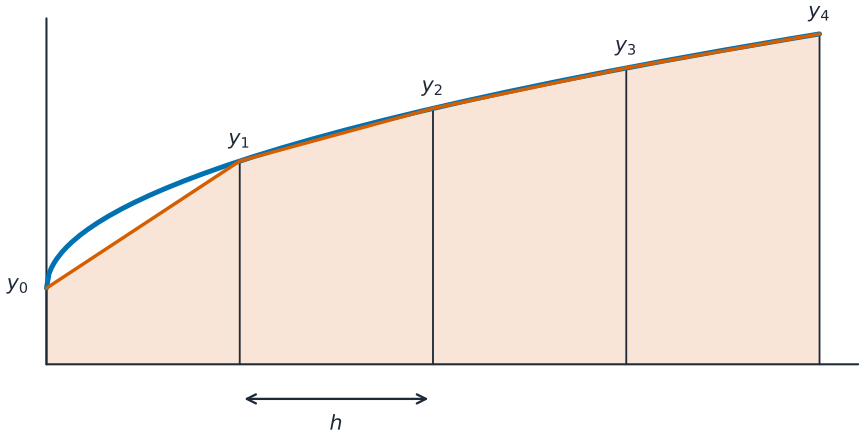
Integration is **reverse differentiation** – reverse each new derivative; harder integrals may need **integration by substitution** (developed in Pure 3). For a linear inside function $(ax + b)$:

$$\int e^{ax+b}\,dx = \frac{1}{a}e^{ax+b} + C, \qquad \int \frac{1}{ax+b}\,dx = \frac{1}{a}\ln|ax+b| + C,$$

$$\int \sin(ax+b)\,dx = -\frac{1}{a}\cos(ax+b) + C, \quad \int \cos(ax+b)\,dx = \frac{1}{a}\sin(ax+b) + C, \quad \int \sec^2(ax+b)\,dx = \frac{1}{a}\tan(ax+b) + C.$$

To integrate a power of sin or cos, first use an identity to remove the power. When you cannot integrate exactly, the **trapezium rule** estimates a definite integral:

$$\int_a^b y\,dx \approx \tfrac{h}{2}\big[y_0 + y_n + 2(y_1 + y_2 + \cdots + y_{n-1})\big].$$



$$\int_a^b y\,dx \approx \tfrac{h}{2}(y_0 + 2(y_1 + \cdots + y_{n-1}) + y_n)$$

Each strip of width h is a trapezium; their areas add up to estimate the integral.

Worked example. Find $\int 6\sin^2 x\,dx$.

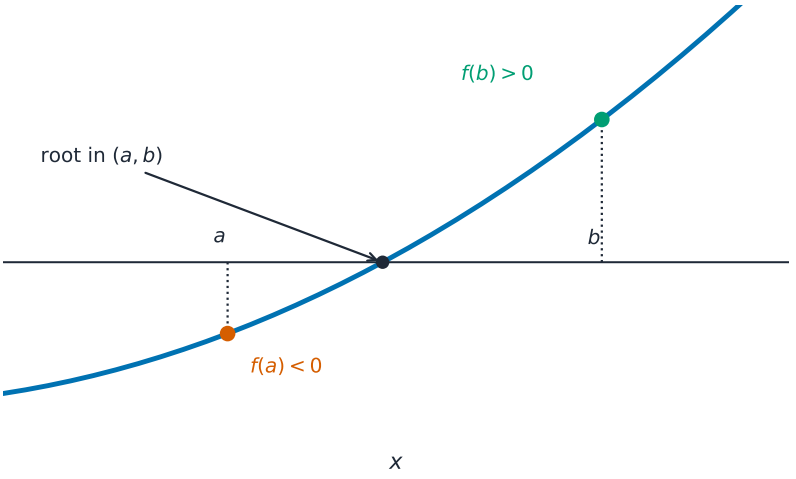
Use $\sin^2 x = \frac{1}{2}(1 - \cos 2x)$:

$$\int 6\sin^2 x\,dx = \int (3 - 3\cos 2x)\,dx = 3x - \tfrac{3}{2}\sin 2x + C.$$

Numerical solution of equations

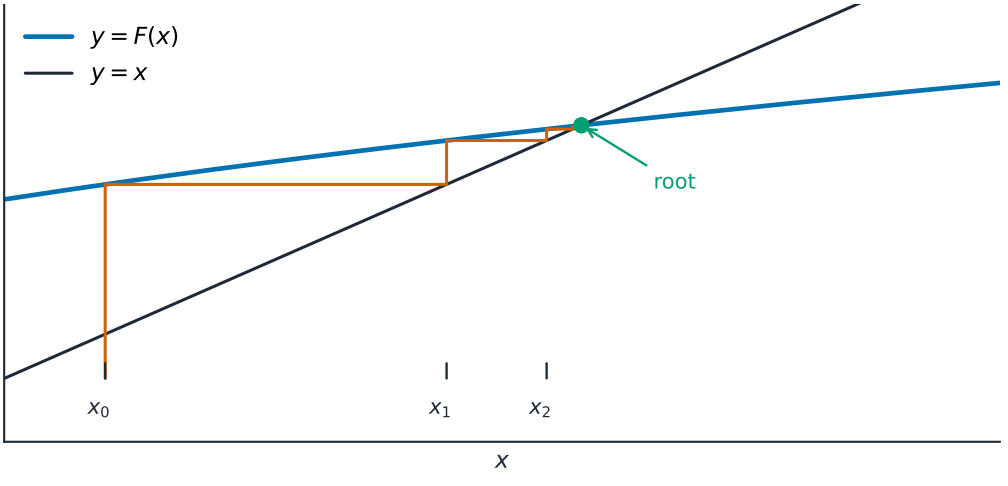
Many equations cannot be solved exactly. Two ideas help you find a **root** (a solution).

- **Sign change**: if $f(a)$ and $f(b)$ have opposite signs (and the graph has no break between them), a root lies between a and b.
- **Iteration**: rearrange the equation into the form $x = F(x)$, then use the **iterative formula** $x_{n+1} = F(x_n)$. Start from a first guess x_0 and repeat. If the values are **convergent** they settle down and **converge** to a root. Keep going until the answer is steady to the accuracy asked for.



if f continuous on $[a, b]$ and $f(a)f(b) < 0$, then a root lies in (a, b)

$f(a)$ and $f(b)$ have opposite signs, so a root is trapped between a and b.



$x_{n+1} = g(x_n)$ · converges near root if $|g'(x)| < 1$

Each step goes up to $y = F(x)$ then across to $y = x$; the staircase closes in on the root.

Worked example. A root β of an equation satisfies $x = \sqrt[3]{-2x - 4.5}$, and $-1.4 < \beta < -1.0$. Use the iteration $x_{n+1} = \sqrt[3]{-2x_n - 4.5}$ with $x_0 = -1.2$.

$x_1 = \sqrt[3]{-2(-1.2) - 4.5} = \sqrt[3]{-2.1} = -1.281, \qquad x_2 = \sqrt[3]{-2(-1.281) - 4.5} = -1.247, \quad \ldots$

The values settle near -1.26, so $\beta = -1.26$ (3 s.f.).

Exam tips

- Use the **laws of logarithms** to solve equations; remember ln and e^x are inverses.
- For **numerical methods**, show a **sign change** to locate a root, set out the iteration clearly, and give the answer to the stated accuracy.
- Learn the **chain, product and quotient rules** and identify which the function needs.
- Solve modulus equations $|f(x)| = g(x)$ by considering both the positive and negative cases, and sketch to check.

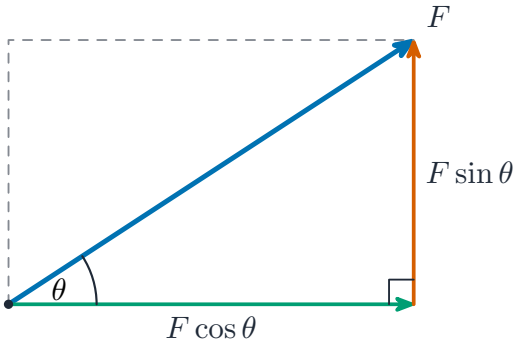
Mechanics

A-Level Mathematics

This handout covers Topic 4: **Mechanics** . It studies how forces make objects move. Throughout this topic, take the acceleration of free fall as $g = 10$ m s^{-2}.

Forces and equilibrium

A **force** is a push or a pull. It is a vector, so it has size and direction. Because it is a vector, you can split a force into **components** (usually horizontal and vertical), and you can add several forces into one **resultant** .



A force at angle θ has a horizontal part $F\cos\theta$ and a vertical part $F\sin\theta$.

A particle is in **equilibrium** when the forces are balanced: the vector sum of the forces is zero. In practice this means the components in any direction add to zero.



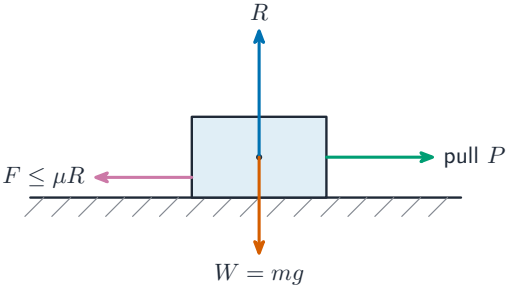
Every mechanics problem is a picture like this. Three forces act on the climber – weight straight down, tension along the rope, and a push from the rock – and because they balance, the climber hangs still in equilibrium. Resolving each into horizontal and vertical parts turns the picture into equations

Friction

When two surfaces touch, the **contact force** between them has two parts: the **normal reaction** R, at right angles to the surface, and the **friction** F, along the surface, which opposes sliding. A "smooth" surface is a model with no friction.

Friction can only grow up to a maximum. At that maximum the body is in **limiting equilibrium**, about to slip, and the friction is **limiting friction**. The maximum is set by the **coefficient of friction** μ:

$$F \leqslant \mu R, \qquad \text{with } F = \mu R \text{ at the point of slipping.}$$



On a rough surface the contact force splits into the normal reaction R and friction F (at most μR).

By **Newton's third law** , the two surfaces push on each other with equal and opposite forces.

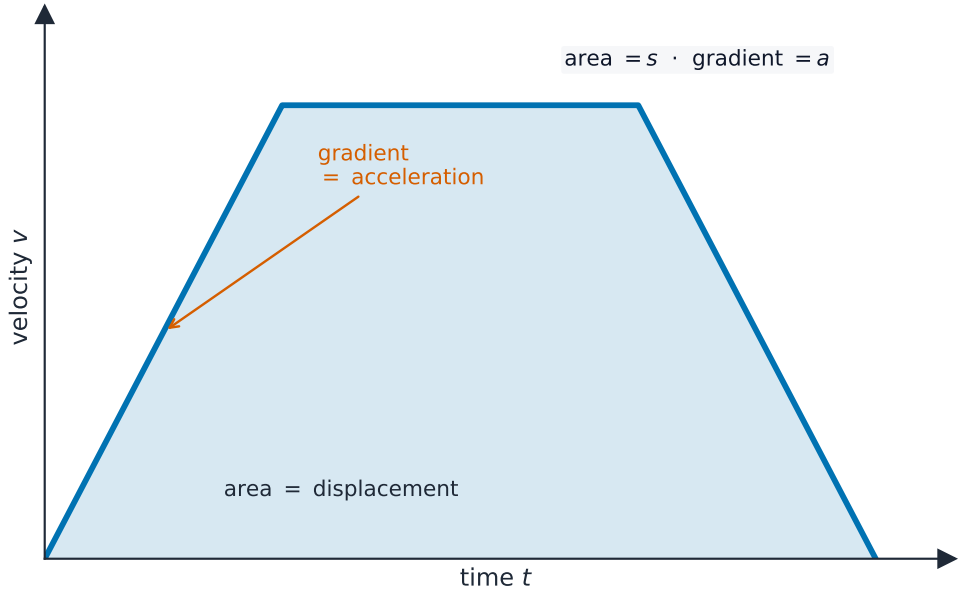
> **Worked example.** A block of weight 20 N rests on a rough horizontal table with coefficient of friction $\mu = 0.4$. Find the largest horizontal force that can be applied before the block slides.

The table's normal reaction balances the weight, so $R = 20$ N. The block is on the point of slipping when friction reaches its maximum $F = \mu R = 0.4 \times 20 = 8$ N. In equilibrium the applied force equals the friction, so the largest force it can resist is 8 N.

Kinematics of motion in a straight line

Distance and **speed** are scalars (size only). **Displacement** , **velocity** and **acceleration** are vectors (size and direction).

On a **velocity-time graph** , the area under the graph is the displacement and the gradient is the acceleration (a negative acceleration is a **deceleration**). On a displacement-time graph, the gradient is the velocity. More generally, differentiate with respect to time to go from displacement to velocity to acceleration, and integrate to go back.



The shaded area gives the distance travelled; the slope of the line gives the acceleration.

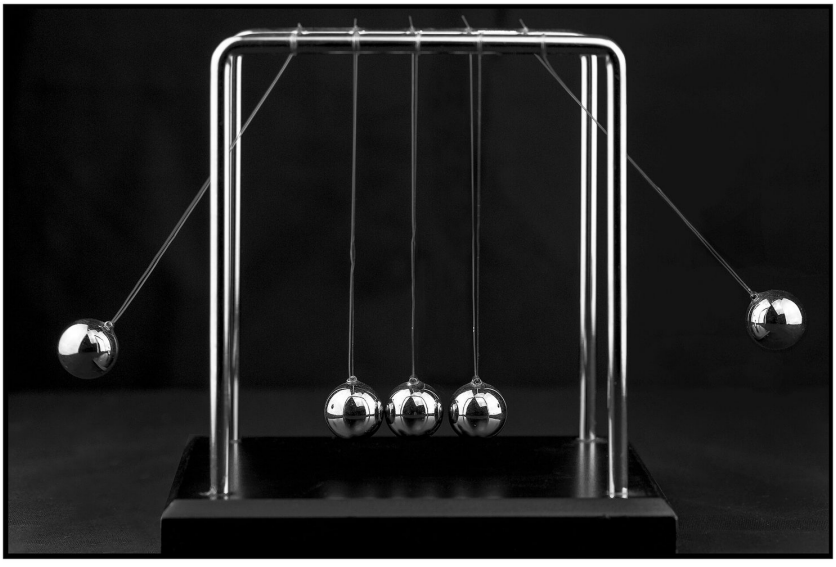
For motion with **constant acceleration** , use these formulae (the "suvat" equations):

$$v = u + at, \qquad s = ut + \tfrac{1}{2}at^2, \qquad v^2 = u^2 + 2as, \qquad s = \tfrac{1}{2}(u + v)t.$$

Worked example. A car starts from rest and accelerates at 2.5 m s^{-2} for 4 s. Find its speed and the distance travelled.

$$v = 0 + 2.5 \times 4 = 10 \text{ m s}^{-1}, \qquad s = 0 + \tfrac{1}{2}(2.5)(4^2) = 20 \text{ m}.$$

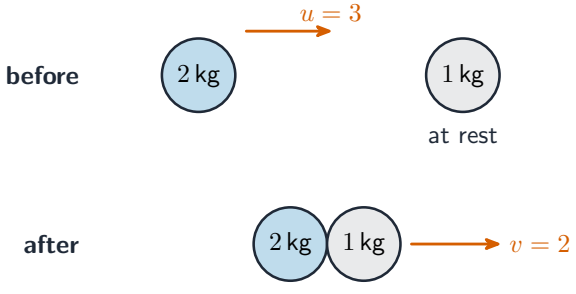
Momentum



A Newton's cradle demonstrates conservation of momentum in collisions.

The **linear momentum** of a body is mass × velocity. It is a vector. In a direct collision of two bodies, the total momentum is unchanged. This is the **conservation of linear momentum** :

$$m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2.$$



$$2(3) + 1(0) = (2 + 1)\,v$$

The total momentum is the same before and after the collision.

Worked example. A body of mass 2 kg moving at 3 m s^{-1} hits a stationary body of mass 1 kg, and they stick together. Find their common speed afterwards.

$$2(3) + 1(0) = (2+1)v \;\Rightarrow\; v = \frac{6}{3} = 2 \text{ m s}^{-1}.$$

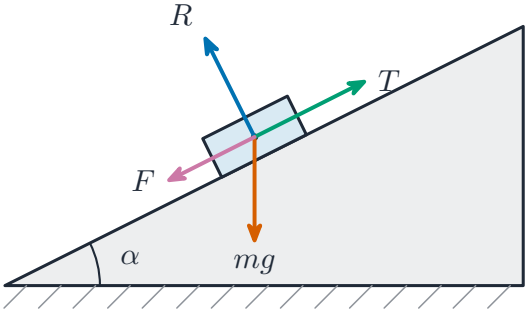
Newton's laws of motion

Newton's laws of motion connect force and acceleration. The key one is: resultant force = **mass** × acceleration,

$$F = ma.$$

The **weight** of a body is the force of gravity on it: $W = mg$. Forces in a problem may include weight, friction, **tension** in a string, the **thrust** (push) in a rod, and **air resistance**.

For motion on an **inclined plane**, split each force into a part along the slope and a part at right angles to it, then apply $F = ma$ along the slope. For **connected particles** (joined by a string), apply $F = ma$ to each body, or to the whole system.



On a slope, resolve the forces along the slope and at right angles to it.

Worked example. A block of mass 12 kg is pulled up a rough plane by a rope parallel to the slope. The plane is at 20° to the horizontal, the coefficient of friction is 0.4, and the acceleration is 2 m s^{-2}. Find the tension in the rope.

The normal reaction is $R = mg\cos 20° = 120\cos 20° = 112.8$ N, so the friction is $F = \mu R = 0.4 \times 112.8 = 45.1$ N. Along the slope, $T - mg\sin 20° - F = ma$:

$$T = ma + mg\sin 20° + F = 12(2) + 120\sin 20° + 45.1 = 24 + 41.0 + 45.1 = 110 \text{ N (3 s.f.)}.$$

Energy, work and power



A roller coaster trades potential energy for kinetic energy as it rises and falls.

The **work done** by a constant force is the **scalar product** of force and displacement – the force times the distance moved in the direction of the force: $W = Fd\cos\theta$, where θ is the angle between the force and the motion. Work is measured in joules (J).

Energy comes in forms you can calculate:

- **Kinetic energy** (energy of movement): $\text{KE} = \frac{1}{2}mv^2$.
- **Gravitational potential energy** (energy of height): $\text{PE} = mgh$.

The work done by the outside forces equals the change in the total energy. When no friction acts, the total energy stays the same – the **conservation of energy**

.

Power is the rate of doing work. For a force pulling in the direction of motion, $P = Fv$ (power = force × velocity). Power is measured in watts (W).

> **Worked example.** A car engine works at 12 kW while the car moves at 20 m s^{-1} on a level road. Find the driving force.
>
> $$P = Fv \;\Rightarrow\; F = \frac{P}{v} = \frac{12000}{20} = 600 \text{ N}.$$

Exam tips

- Draw a clear **force diagram** and resolve into perpendicular components; for equilibrium, each direction sums to zero.
- Use **SUVAT** only for constant acceleration and keep a consistent positive direction.
- Apply $F = ma$ along the direction of motion, including friction ($F = \mu R$) on a rough surface.
- State your **assumptions** (light inextensible string, smooth pulley, particle) – they are often worth a mark.

Probability & Statistics 1

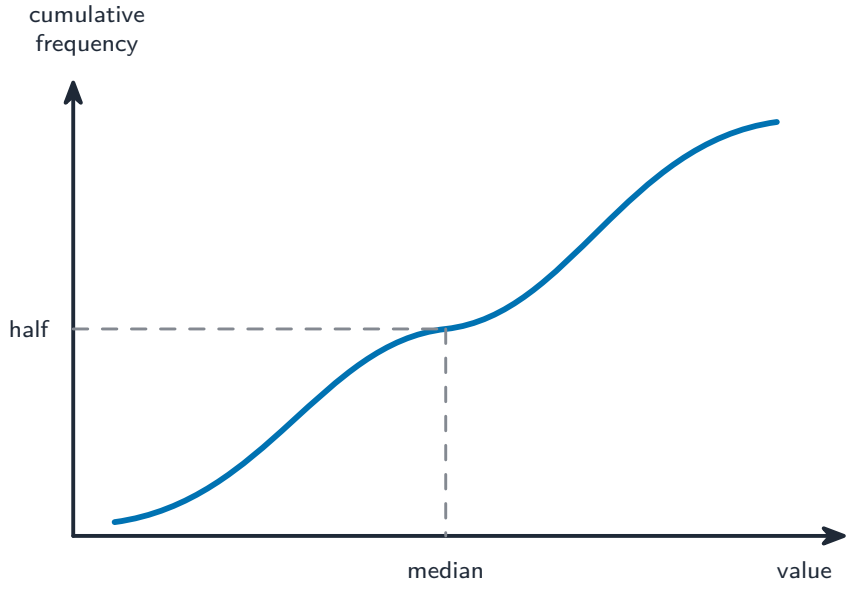
A-Level Mathematics

This handout covers Topic 5: **Probability & Statistics** 1. It is about describing data, counting choices, and working out the chance of events.

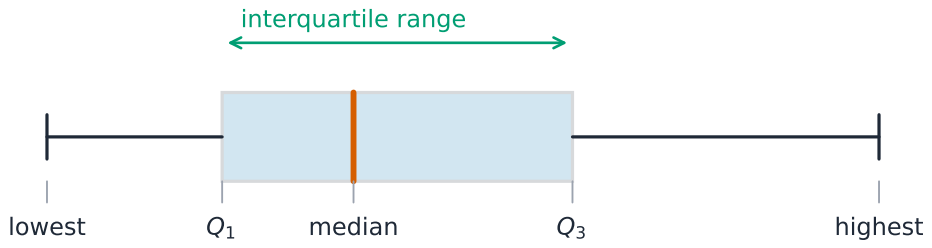
Representation of data

Choose a diagram that suits the data. You should be able to draw and read:

- a **stem-and-leaf diagram** (keeps the original values and shows the shape); two data sets are compared with a **back-to-back** version – a shared central stem, one set's leaves increasing to the left and the other's to the right, so you can compare their medians and spreads at a glance;
- a **box-and-whisker plot** (shows the lowest value, the three quartiles, and the highest value);
- a **histogram** (for grouped data, where the **area** of each bar shows the frequency);
- a **cumulative frequency** graph (running totals, used to estimate the median and quartiles).



A cumulative frequency curve is S-shaped; read the median across from half the total frequency, and the quartiles from one and three quarters



box: Q_1–Q_3 · median line · whiskers: min/max

The box spans the quartiles Q_1 to Q_3; the whiskers reach the lowest and highest values.

■ Averages and spread

A **measure of central tendency** is a single "middle" value:

- the **mean** $\bar{x} = \dfrac{\sum x}{n}$ (the average);
- the **median** (the middle value when the data is in order);
- the **mode** (the most common value).

A measure of **variation** shows how spread out the data is:

- the **range (of data)** (highest − lowest);
- the **interquartile range** (upper quartile − lower quartile);
- the **standard deviation** $\sigma = \sqrt{\dfrac{\sum x^2}{n} - \bar{x}^2}$.

You often work from the totals $\sum x$ and $\sum x^2$. The square of the standard deviation is the **variance** .

> **Worked example.** For 10 values, $\sum x = 50$ and $\sum x^2 = 300$. Find the mean and standard deviation.
>
> $$\bar{x} = \frac{50}{10} = 5, \qquad \sigma = \sqrt{\frac{300}{10} - 5^2} = \sqrt{30 - 25} = \sqrt{5} = 2.24.$$

Coding makes big numbers easier. Replace each value by $t = x - a$ for a convenient **assumed mean** a. Then $\bar{x} = a + \bar{t}$, while the standard deviation is **unchanged** (shifting every value does not spread the data). So from the coded totals $\sum(x-a)$ and $\sum(x-a)^2$ you get $\bar{x}$ and σ directly, and two data sets can be compared through their coded totals.

Permutations and combinations

A **permutation** is an arrangement where order matters; a **combination** is a selection where order does not matter. The numbers are

$$ {}^{n}P_{r} = \frac{n!}{(n-r)!}, \qquad {}^{n}C_{r} = \binom{n}{r} = \frac{n!}{r!\,(n-r)!}. $$

To arrange objects in a line when some are repeated, divide by the factorial of each repeat count.

permutation (order matters)

AB BA

combination (order does not)

$\{A, B\}$

Order matters for a permutation, but not for a combination

> **Worked example.** How many different arrangements are there of the letters of the word NEEDLESS?

There are 8 letters, with E repeated 3 times and S repeated 2 times:

$$ \frac{8!}{3!\,2!} = \frac{40320}{6 \times 2} = 3360. $$

Arranging 3 of 4 objects ($^4P_3$)

4	×	3	×	2	= 24
1st pick		2nd pick		3rd pick	

Choosing 3 of 4 (order ignored): $^4C_3 = \frac{^4P_3}{3!} = 4$

$$^nP_r = n!/(n-r)! \quad \cdot \quad ^nC_r = {}^nP_r/r!$$

$$nPr = n!/(n-r)!;\ nCr = nPr/r!$$

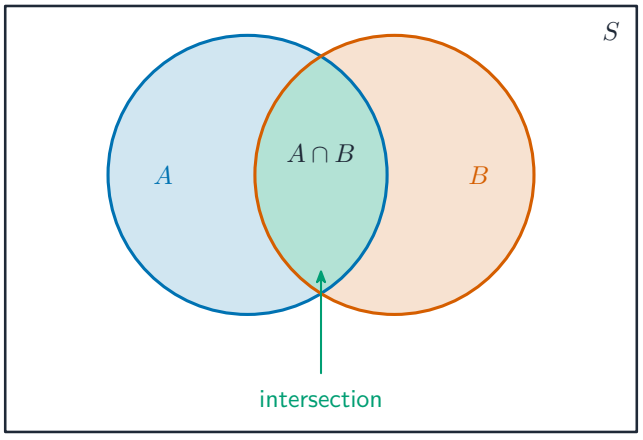
Probability



Dice: a familiar starting point for probability.

Find a probability by counting equally likely outcomes, or by using permutations and combinations. Combine probabilities with these rules:

- **addition** for "or": $P(A \cup B) = P(A) + P(B) - P(A \cap B)$;
- **multiplication** for "and" when events are independent: $P(A \cap B) = P(A)\,P(B)$.



The overlap of the two circles is $A \cap B$; the addition rule subtracts it once so it is not counted twice.

Two events are **mutually exclusive events** if they cannot both happen, and **independent events** if one happening does not change the chance of the other. To test independence, check whether $P(A \cap B) = P(A) \times P(B)$. A **conditional probability** is the chance of A given that B has happened: $P(A \mid B) = \dfrac{P(A \cap B)}{P(B)}$.

> **Worked example.** Events have $P(A) = 0.5$, $P(B) = 0.4$ and $P(A \cap B) = 0.2$. Are A and B independent?

Test: $P(A) \times P(B) = 0.5 \times 0.4 = 0.2 = P(A \cap B)$. The values are equal, so A and B are independent.

Discrete random variables

A **discrete random variable** X takes separate values, each with a probability. List them in a **probability distribution table**; the probabilities must add to 1. Then the **expectation** (mean) and **variance** are

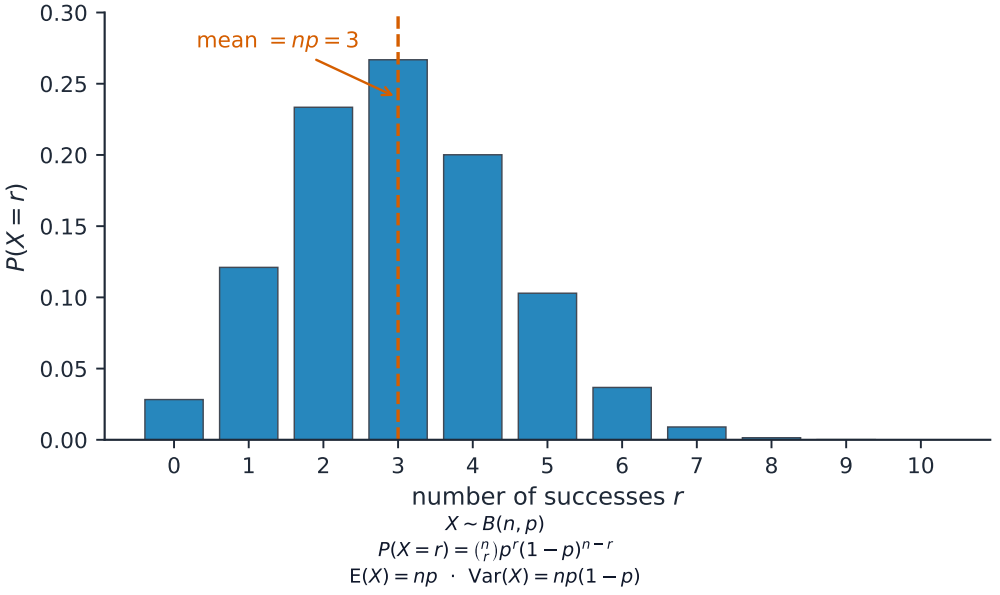
$$E(X) = \sum x\,P(X = x), \qquad \operatorname{Var}(X) = \sum x^2\,P(X = x) - \big(E(X)\big)^2.$$

Two special models:

- the **binomial distribution** $X \sim B(n, p)$, for the number of successes in n independent trials: $P(X = r) = \binom{n}{r} p^r (1-p)^{n-r}$, with $E(X) = np$ and $\operatorname{Var}(X) = np(1-p)$;
- the **geometric distribution**, for the trial on which the first success happens: $P(X = r) = (1-p)^{r-1}p$, with $E(X) = \dfrac{1}{p}$.

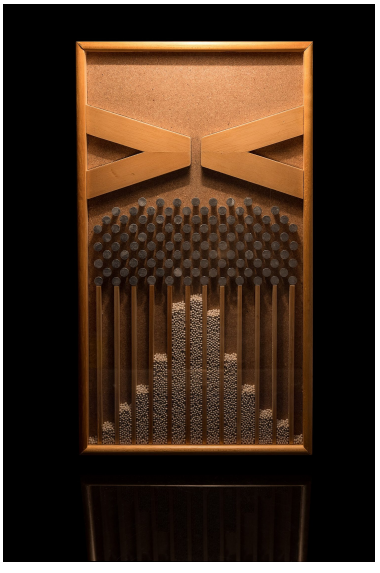
Worked example. $X \sim B(10, 0.3)$. Find $P(X = 2)$ and $E(X)$.

$$P(X = 2) = \binom{10}{2}(0.3)^2(0.7)^8 = 45 \times 0.09 \times 0.05765 = 0.233, \qquad E(X) = 10 \times 0.3 = 3.$$



The distribution $B(10, 0.3)$: each bar is $P(X = r)$, clustered around the mean $np = 3$.

The normal distribution

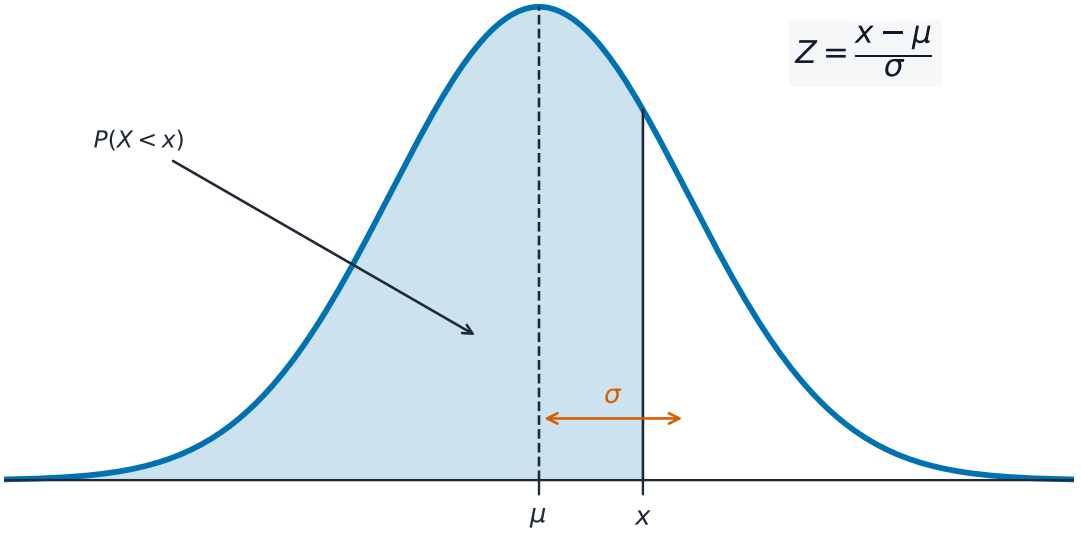


A Galton board: balls falling through pins pile up into the bell-shaped normal distribution.

The **normal distribution** models a **continuous random variable** with a symmetric bell shape. Write $X \sim N(\mu, \sigma^2)$, where μ is the mean and σ is the standard deviation. To use the tables, by **standardisation** convert to the variable $Z \sim N(0, 1)$:

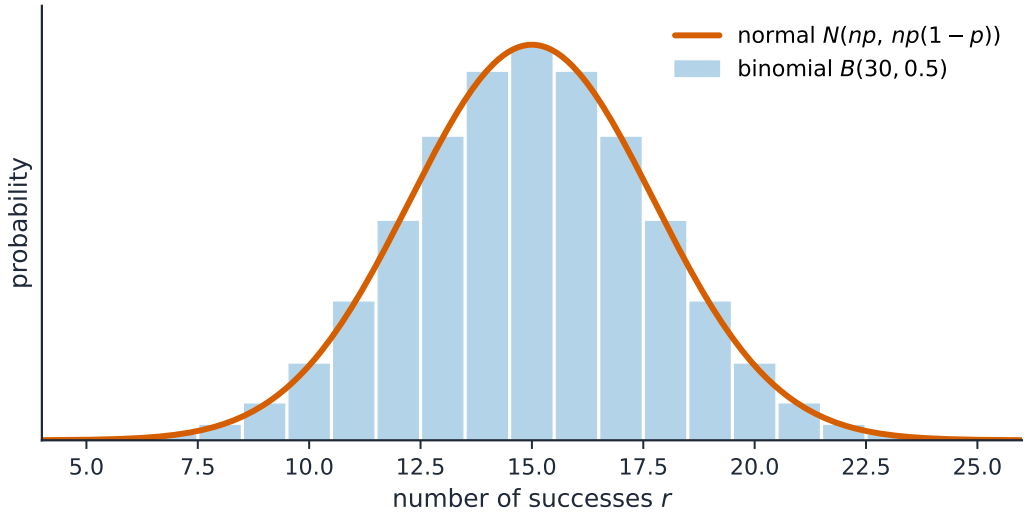
$$Z = \frac{X - \mu}{\sigma}.$$

Then $P(X < x) = P\left(Z < \frac{x - \mu}{\sigma}\right)$, which you read from the normal table Φ.



A normal probability is an area under the bell curve; standardizing rescales it to $Z \sim N(0, 1)$.

The normal distribution is also a good **approximation** to the binomial when n is large. Because you replace a discrete variable by a continuous one, apply a **continuity correction** (adjust by 0.5).



approx $N(np,\ np(1-p))$ when n large · continuity correction

When n is large the binomial bars follow a normal curve of the same mean and variance.

Worked example. Bags of rice have mass $X \sim N(\mu, 0.14^2)$. Given that $P(X < 1.48) = 0.22$, find μ.

From the table, $P(Z < z) = 0.22$ gives $z = -0.772$. So

$$\frac{1.48 - \mu}{0.14} = -0.772 \;\Rightarrow\; \mu = 1.48 + 0.772 \times 0.14 = 1.59 \text{ kg (3 s.f.)}.$$

Exam tips

- Decide whether order matters: **permutations** $({}^nP_r)$ when it does, **combinations** $({}^nC_r)$ when it does not.
- For a **discrete random variable**, check the probabilities sum to 1 and use $E(X) = \sum x\,P(X = x)$.
- For the **normal distribution**, standardise with $z = (x-\mu)/\sigma$, **sketch and shade**, then read the table.
- Apply a **continuity correction** when approximating a discrete variable by the normal.

Further Pure Mathematics 1

A-Level Further Mathematics

This handout covers Topic 1: **Further Pure Mathematics** 1. It adds new algebra, matrices, polar coordinates, more vectors, and proof by induction.

Roots of polynomial equations

For a polynomial equation, the **roots** are linked to the **coefficients** . For $ax^2 + bx + c = 0$ with roots α, β:

$$\alpha + \beta = -\frac{b}{a}, \qquad \alpha\beta = \frac{c}{a}.$$

For a cubic $ax^3 + bx^2 + cx + d = 0$ with roots α, β, γ:

$$\sum \alpha = -\frac{b}{a}, \qquad \sum \alpha\beta = \frac{c}{a}, \qquad \alpha\beta\gamma = -\frac{d}{a}.$$

Sums like $\sum \alpha$ and $\sum \alpha\beta$ are **symmetric functions** of the roots (unchanged if the roots are swapped). To find an equation whose roots are changed in a simple way, use a **substitution** (for example, put $w = \alpha + 1$).

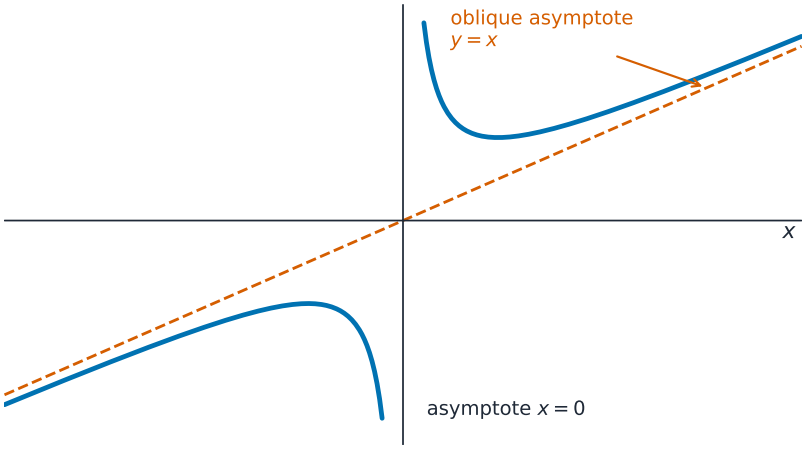
> **Worked example.** The equation $x^2 - 5x + 6 = 0$ has roots α, β. Find the equation with roots $\alpha + 1, \beta + 1$.

Here $\alpha + \beta = 5$ and $\alpha\beta = 6$. The new sum is $(\alpha + 1) + (\beta + 1) = 7$, and the new product is $(\alpha + 1)(\beta + 1) = \alpha\beta + \alpha + \beta + 1 = 6 + 5 + 1 = 12$. So the new equation is

$$x^2 - 7x + 12 = 0.$$

Rational functions and graphs

A **rational function** is a fraction of two polynomials. When the top has degree exactly one higher than the bottom, the graph has an **oblique asymptote** (a slanted line the curve approaches); find it by dividing out. You should also be able to relate the graph of $y = f(x)$ to those of $y^2 = f(x)$, $y = \dfrac{1}{f(x)}$, $y = |f(x)|$ and $y = f(|x|)$. Find the set of values the function can take using the **discriminant** , and locate its **turning points** .



oblique asymptote when deg(num) = deg(den) + 1 (long division)

Far from the origin the curve hugs the slant line $y = x$; near $x = 0$ it runs off along the vertical asymptote.

Worked example. Find the oblique asymptote of $y = \dfrac{x^2 + x + 1}{x + 1}$.

Divide out the fraction: $\dfrac{x^2 + x + 1}{x + 1} = x + \dfrac{1}{x + 1}$. As $x \to \pm\infty$ the remainder $\dfrac{1}{x + 1} \to 0$, so the curve approaches the line $y = x$ – the oblique asymptote.

Summation of series

Learn the standard results:

$$\sum_{r=1}^{n} r = \tfrac{1}{2}n(n+1), \qquad \sum_{r=1}^{n} r^2 = \tfrac{1}{6}n(n+1)(2n+1), \qquad \sum_{r=1}^{n} r^3 = \tfrac{1}{4}n^2(n+1)^2.$$

The **method of differences** sums a series by cancelling middle terms: if **partial fractions** write each term as $f(r) - f(r+1)$, almost everything cancels. From the sum to n terms you can see whether a series is **convergent** and, if so, find its **sum to infinity** .

Worked example. Find $\displaystyle\sum_{r=1}^{n}(2r + 1)$.

$$\sum_{r=1}^{n}(2r+1) = 2\sum_{r=1}^{n} r + \sum_{r=1}^{n} 1 = 2 \cdot \tfrac{1}{2}n(n+1) + n = n(n+1) + n = n(n+2).$$

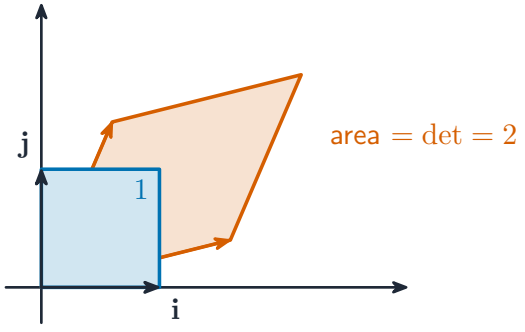
Matrices

A **matrix** is a rectangular block of numbers. **Matrix addition** adds entries in matching positions; you can also subtract and multiply matrices (multiplication is row × column). The **zero matrix** has every entry 0, and the **identity matrix** (or **unit matrix**) I leaves any matrix unchanged under multiplication.

For a 2×2 matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, the **determinant** is $\det A = ad - bc$. If $\det A \neq 0$ the matrix is non-singular (otherwise it is a **singular matrix**), and the **inverse matrix** is

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

For a product, $(AB)^{-1} = B^{-1}A^{-1}$. A 2×2 matrix can represent a **geometric transformation** of the plane – a **rotation**, **reflection**, **enlargement**, **stretch** or **shear**: the determinant gives the area scale factor, and a product AB means "do B, then A". Points or lines that do not move are called **invariant points** and **invariant lines**.



The matrix sends the unit square to a parallelogram whose area is the determinant.

Worked example. Find the inverse of $A = \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}$.

$\det A = 3 \times 4 - 1 \times 2 = 10$, so

$$A^{-1} = \frac{1}{10} \begin{pmatrix} 4 & -1 \\ -2 & 3 \end{pmatrix}.$$

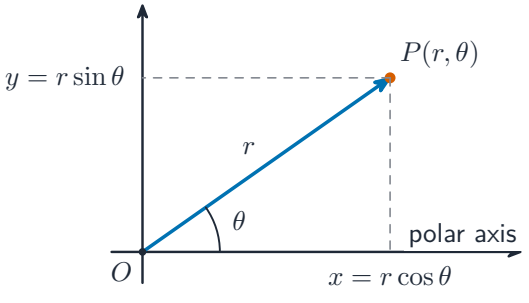
Polar coordinates



A spiral staircase: spirals are described naturally using polar coordinates.

Polar coordinates give a point by its distance r from the origin and its angle θ. They link to **Cartesian coordinates** by

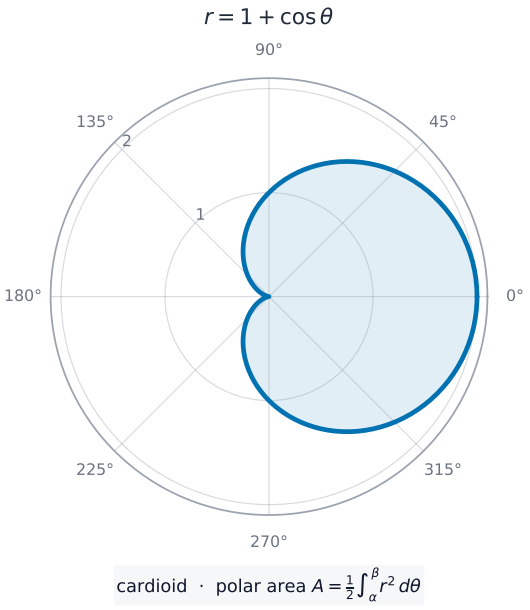
$$x = r\cos\theta, \qquad y = r\sin\theta, \qquad r^2 = x^2 + y^2.$$



Polar coordinates (r,θ) convert to Cartesian by $x = r\cos\theta$ and $y = r\sin\theta$.

You should sketch simple **polar curves** , and find the area of a sector with

$$\text{area} = \tfrac{1}{2}\int r^2\,d\theta.$$



The cardioid $r = 1 + \cos\theta$, plotted straight from r as a function of θ.

Worked example. Convert the polar equation $r = 4\cos\theta$ to Cartesian form.

Multiply both sides by r: $r^2 = 4r\cos\theta$, so $x^2 + y^2 = 4x$. Completing the square gives $(x-2)^2 + y^2 = 4$, a circle of radius 2 centred at $(2, 0)$.

Vectors



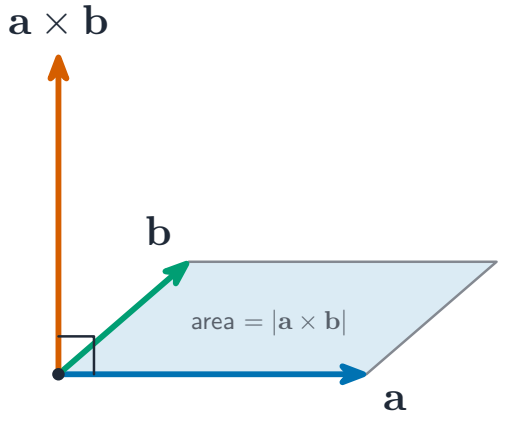
Forces like wind and water are vectors – they have both size and direction.

In three dimensions a **plane** can be written as $ax + by + cz = d$, or $\mathbf{r}\cdot\mathbf{n} = p$, or

$\mathbf{r} = \mathbf{a} + \lambda\mathbf{b} + \mu\mathbf{c}$. Besides the scalar product, there is the **vector product** of two **vectors** :

$$\mathbf{a} \times \mathbf{b} = |\mathbf{a}|\,|\mathbf{b}| \sin\theta\, \hat{\mathbf{n}},$$

which gives a vector at right angles to both. With scalar and vector products you can find distances, angles, and where lines and planes meet – including the shortest distance between **skew lines** .



$\mathbf{a} \times \mathbf{b}$ *is perpendicular to both vectors; its length equals the area of the parallelogram they span.*

Worked example. Find $\mathbf{a} \times \mathbf{b}$ for $\mathbf{a} = \mathbf{i} + \mathbf{j}$ and $\mathbf{b} = \mathbf{j} + \mathbf{k}$.

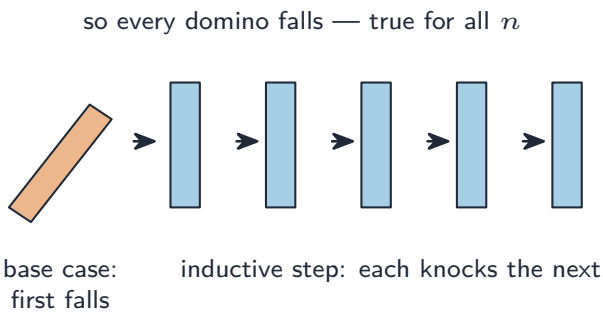
$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} = \mathbf{i}(1) - \mathbf{j}(1) + \mathbf{k}(1) = \mathbf{i} - \mathbf{j} + \mathbf{k}.$$

Proof by induction

Mathematical induction proves a result for every positive integer n in two steps:

1. **Base case**: show the result is true for $n = 1$.
2. **Inductive step**: assume it is true for $n = k$, then prove it for $n = k + 1$.

If both steps work, the result is true for all n. Often you first make a **conjecture** (a sensible guess) from a few cases, then confirm it by **inductive proof** .



Induction is like dominoes: the base case topples the first, and the inductive step makes each one knock over the next – so the result holds for every n

Worked example. Prove that $\sum_{r=1}^{n} r = \frac{1}{2}n(n+1)$.

Base case $n = 1$: the left side is 1 and the right side is $\frac{1}{2}(1)(2) = 1$. True. **Inductive step**: assume $\sum_{r=1}^{k} r = \frac{1}{2}k(k+1)$. Then

$$\sum_{r=1}^{k+1} r = \tfrac{1}{2}k(k+1) + (k+1) = (k+1)\left(\tfrac{k}{2}+1\right) = \tfrac{1}{2}(k+1)(k+2).$$

This is the formula with $n = k+1$, so by induction it holds for all n.

Exam tips

- Use the **relationships between roots and coefficients** (sum, sum of pairs, product) instead of solving the polynomial.
- For **proof by induction**, set out the base case, assume the result for $n = k$, prove it for $n = k+1$, and write a clear closing statement.
- Sketch **rational-function graphs** with their asymptotes and turning points; use the **method of differences** for a summation.
- For **matrices**, know the determinant, inverse and invariant lines, and interpret each transformation geometrically.

Further Mechanics

A-Level Further Mathematics

This handout covers Topic 3: **Further Mechanics** . It extends mechanics to projectiles, rigid bodies, circular motion, elastic strings, variable forces and collisions. Take $g = 10$ m s^{-2}.

Motion of a projectile

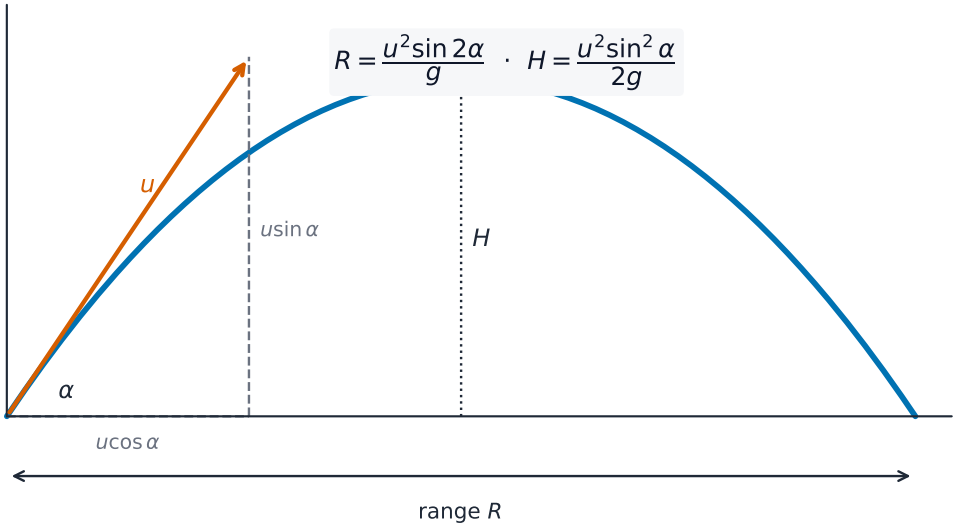


Water jets follow parabolic paths – the classic projectile motion under gravity.

A **projectile** moves freely under gravity, so it has **constant acceleration** g downwards and no horizontal acceleration. Treat the horizontal and vertical motions separately. If it is launched at speed u and angle α:

$$\text{horizontal: } x = u\cos\alpha \cdot t, \qquad \text{vertical: } y = u\sin\alpha \cdot t - \tfrac{1}{2}gt^2.$$

Eliminating t gives the **Cartesian equation** of the **trajectory** (the path), which is a parabola. For a fixed launch speed, all the trajectories lie under a **bounding parabola** (the envelope of reachable points). The range on level ground is $\dfrac{u^2\sin 2\alpha}{g}$ and the greatest height is $\dfrac{u^2\sin^2\alpha}{2g}$.



The path is a parabola; the launch speed u splits into a steady horizontal part and a vertical part slowed by gravity.

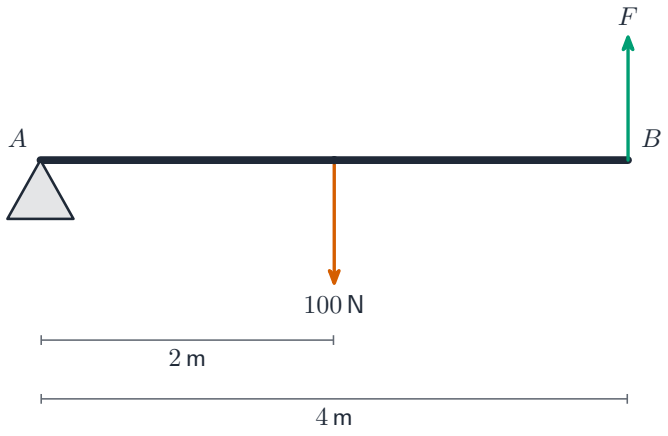
Worked example. A ball is thrown at $u = 20$ m s^{-1} at $30°$ to the horizontal. Find the range and greatest height.

$$\text{range} = \frac{20^2 \sin 60°}{10} = 40 \times 0.866 = 34.6 \text{ m}, \qquad \text{height} = \frac{20^2 \sin^2 30°}{2 \times 10} = \frac{400 \times 0.25}{20} = 5 \text{ m}.$$

Equilibrium of a rigid body

The **moment** of a force about a point is force × perpendicular distance; it measures turning effect. The weight of a body acts at its **centre of mass**, which you can find using **symmetry** for a uniform flat shape (a **lamina**), or by treating a composite body as a set of particles.

A rigid body under **coplanar forces** is in **equilibrium** when two conditions both hold: the vector sum of the forces is zero, and the sum of the moments about any point is zero. A body may also be on the edge of **toppling** (turning over) or **sliding** (slipping).



Taking moments about the pivot A balances the turning effects: $F \times 4 = 100 \times 2$.

> **Worked example.** A uniform beam AB of length 4 m and weight 100 N rests on a pivot at A. A vertical force F at B keeps it horizontal. Find F.

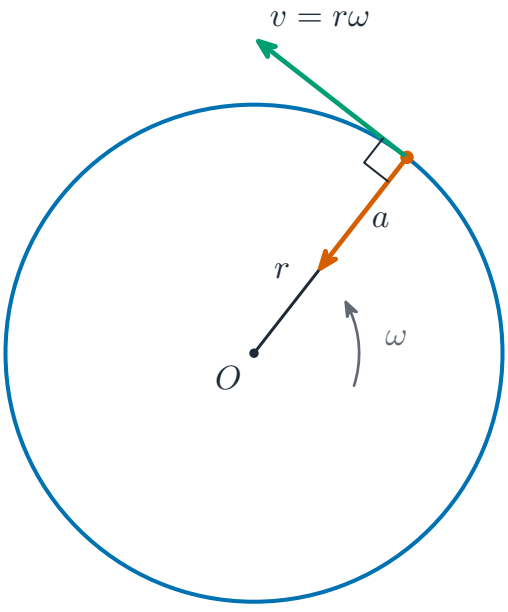
Take moments about A (the weight acts at the centre, 2 m from A):

$$F \times 4 = 100 \times 2 \;\Rightarrow\; F = 50 \text{ N}.$$

Circular motion

For a particle moving in a circle of radius r, the **angular speed** ω links to the speed by $v = r\omega$. The acceleration points towards the centre – the **centripetal acceleration** – with size

$$a = r\omega^2 = \frac{v^2}{r}.$$



The velocity points along the tangent; the acceleration points inward to the centre.



The hammer thrower is a moving diagram of this page. The taut wire pulls the ball toward the centre – that inward pull is the centripetal force – while the ball's velocity points along the tangent. Let go, and with no inward force left the ball flies straight off that tangent

In a **horizontal circle** the speed is constant – as in a **conical pendulum** (a mass swung on a string). In a **vertical circle** use energy conservation, because the speed changes with height; the **normal contact force** or string tension provides the centripetal force.

Worked example. A particle moves in a horizontal circle of radius 2 m with angular speed 3 rad s^{-1}. Find its speed and acceleration.

$$v = r\omega = 2 \times 3 = 6 \text{ m s}^{-1}, \qquad a = r\omega^2 = 2 \times 3^2 = 18 \text{ m s}^{-2}.$$

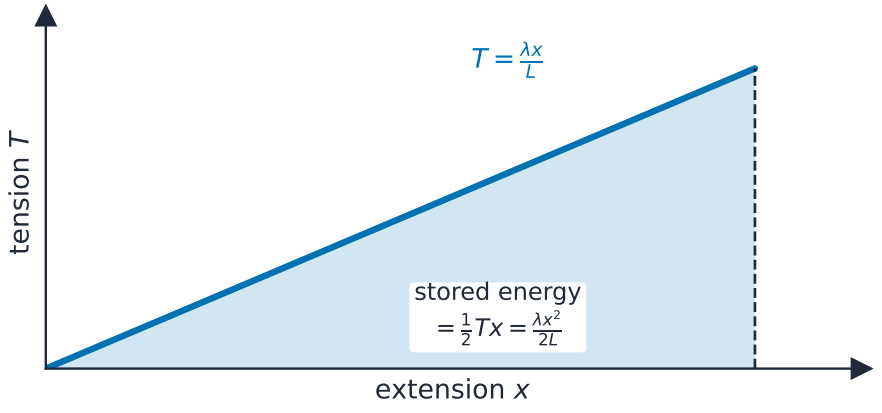
Hooke's law

Hooke's law says the tension in an elastic string or spring is proportional to its extension x:

$$T = \frac{\lambda x}{L},$$

where L is the natural length and λ is the **modulus of elasticity**. Stretching stores **elastic potential energy**:

$$E = \frac{\lambda x^2}{2L}.$$



$T = \frac{\lambda}{L}x$ · $E = \frac{1}{2}Tx = \frac{\lambda x^2}{2L}$ (area under graph)

Tension rises in proportion to extension; the shaded triangle is the stored elastic energy.

Worked example. An elastic string of natural length 2 m and modulus 50 N is stretched by 0.5 m. Find the tension and the stored energy.

$$T = \frac{50 \times 0.5}{2} = 12.5 \text{ N}, \qquad E = \frac{50 \times 0.5^2}{2 \times 2} = 3.125 \text{ J}.$$

Linear motion under a variable force

When the force depends on position x, use acceleration in the form $a = v\dfrac{dv}{dx}$, which turns Newton's law into a **differential equation** relating v and x.

which form of acceleration?

force depends on **time** t	force depends on **position** x
$a = \dfrac{dv}{dt}$	$a = v\,\dfrac{dv}{dx}$
gives v as a function of t	gives v as a function of x

both come from $a = \dfrac{dv}{dt}$ with the chain rule; pick the one matching what the force depends on

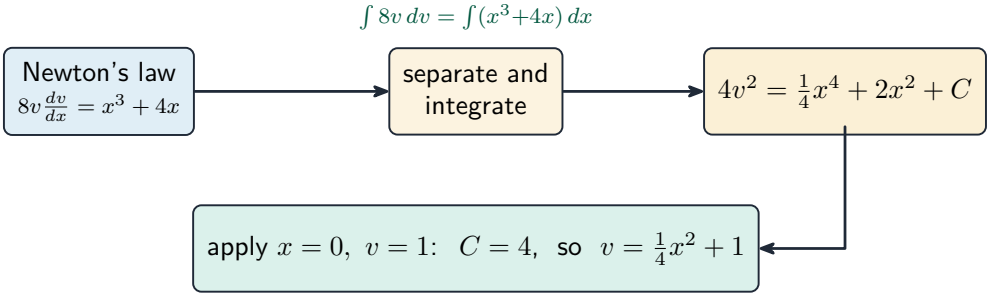
Use dv/dt when the force depends on time, v dv/dx when it depends on position

Worked example. A particle of mass 8 kg moves along a line under a **variable force** of size (x^3+4x) N acting in the direction of motion. When $x = 0$, $v = 1$. Find v in terms of x.

Newton's law gives $8v\dfrac{dv}{dx} = x^3 + 4x$. Separating and integrating:

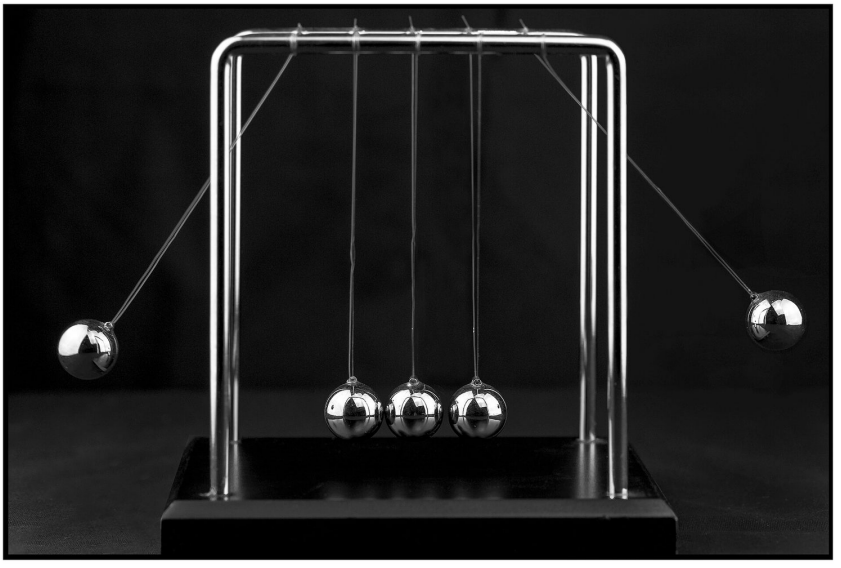
$$\int 8v\,dv = \int (x^3 + 4x)\,dx \;\Rightarrow\; 4v^2 = \tfrac{1}{4}x^4 + 2x^2 + C.$$

At $x = 0$, $v = 1$ gives $C = 4$, so $4v^2 = \tfrac{1}{4}x^4 + 2x^2 + 4 = 4\left(\tfrac{x^2}{4} + 1\right)^2$. Hence $v = \tfrac{1}{4}x^2 + 1$.



a position-dependent force becomes a **differential equation** in v and x

A position-dependent force becomes a differential equation in v and x

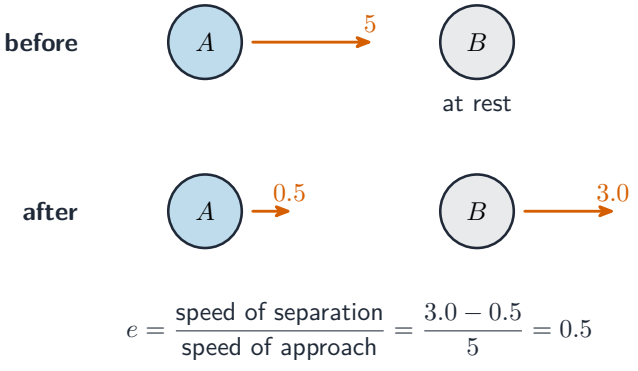


A Newton's cradle demonstrates conservation of momentum in collisions.

In a collision, total **linear momentum** is conserved – the **conservation of linear momentum** . The bounciness is measured by **Newton's experimental law** , which defines the **coefficient of restitution** e:

$$e = \frac{\text{speed of separation}}{\text{speed of approach}}, \qquad 0 \leqslant e \leqslant 1.$$

Here $e = 1$ is perfectly elastic (no energy lost) and $e = 0$ is inelastic (the bodies stay together). In an **oblique impact** , resolve the velocities along and perpendicular to the line of impact, applying restitution along it.



The coefficient of restitution e compares how fast the bodies separate with how fast they approached.

Worked example. A sphere A of mass 2 kg moving at 5 m s^{-1} hits a stationary sphere B of mass 3 kg, with $e = 0.5$. Find the speeds afterwards.

Momentum: $2(5) = 2v_A + 3v_B$, so $2v_A + 3v_B = 10$. Restitution: $v_B - v_A = 0.5(5) = 2.5$. Solving together gives $v_A = 0.5$ m s^{-1} and $v_B = 3.0$ m s^{-1}.

Exam tips

- For **projectiles**, resolve into horizontal (constant velocity) and vertical ($a = g$) motion, linked by the same time.
- For a **rigid body in equilibrium**, take moments about a point that removes an unknown force.
- For **circular motion**, use $F = mv^2/r = m\omega^2 r$ towards the centre; in a vertical circle check the minimum speed at the top.
- With a **variable force**, use $a = v\,\frac{dv}{dx}$ and integrate; elastic PE $= \frac{\lambda x^2}{2L}$.

Further Probability & Statistics

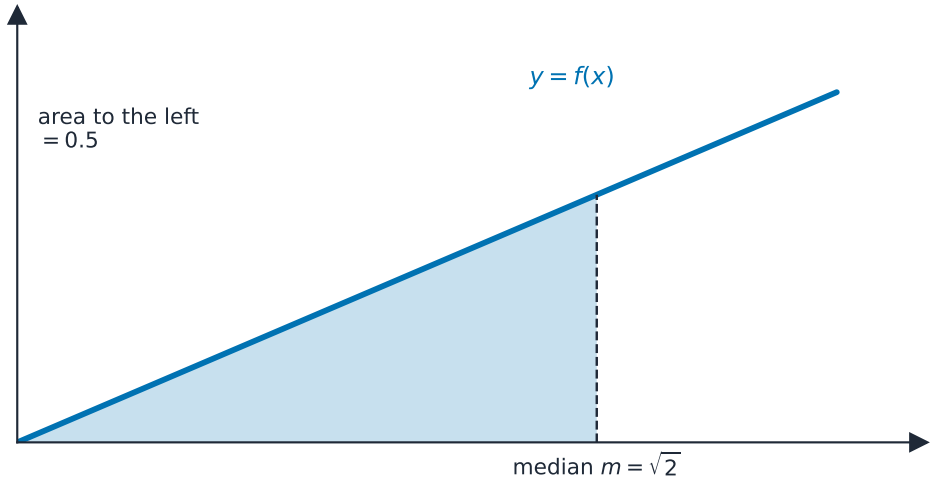
A-Level Further Mathematics

This handout covers Topic 4: **Further Probability & Statistics** . It adds continuous distributions, small-sample inference, the chi-squared and non-parametric tests, and probability generating functions.

Continuous random variables

A continuous variable X is described by a **probability density function** $f(x)$, which may be defined piecewise. The probability over a range is the area under f, and the mean of any function of X is

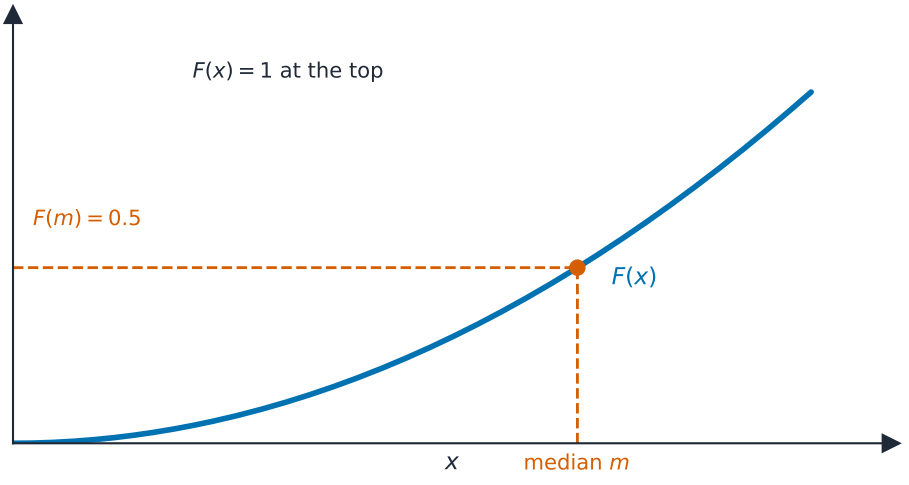
$$E(g(X)) = \int g(x)\, f(x)\, dx.$$



median m solves $F(m) = \int_{-\infty}^{m} f(x)\,dx = \frac{1}{2}$

The median sits where the area under $f(x)$ to its left is exactly 0.5.

The **cumulative distribution function** $F(x) = P(X \leqslant x)$ is the running total: $F(x) = \int_{-\infty}^{x} f(t)\, dt$, and $f(x) = F'(x)$. Use F to find probabilities and **percentiles** (for example, the **median** solves $F(x) = 0.5$).



$F(x) = P(X \leq x) = \int_{-\infty}^{x} f \;\cdot\; 0 \leq F \leq 1 \;\cdot\; \text{median: } F(m) = \frac{1}{2}$

The cumulative graph $F(x)$ rises from 0 to 1; the height 0.5 is reached at the median.

> **Worked example.** A variable has $f(x) = \frac{1}{2}x$ for $0 \leqslant x \leqslant 2$. Find the median.

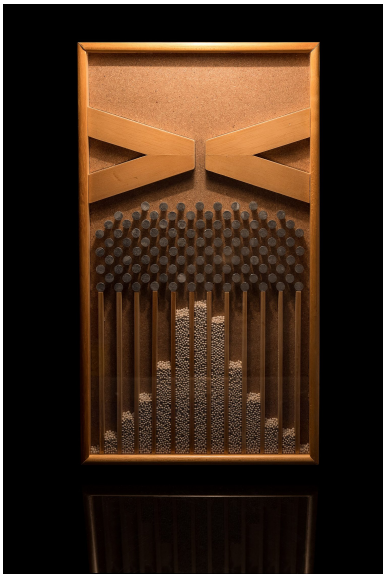
The cumulative distribution function is $F(x) = \displaystyle\int_0^x \tfrac{1}{2}t\,dt = \tfrac{1}{4}x^2$. Set $F(m) = 0.5$:

$$\tfrac{1}{4}m^2 = 0.5 \;\Rightarrow\; m^2 = 2 \;\Rightarrow\; m = \sqrt{2} = 1.41.$$

The distribution of a related variable. If $Y = g(X)$, find Y's distribution through its cumulative function. For $Y = X^2$: $F_Y(y) = P(X^2 \leqslant y) = P(-\sqrt{y} \leqslant X \leqslant \sqrt{y}) = F_X(\sqrt{y}) - F_X(-\sqrt{y})$, then differentiate for $f_Y = F_Y'$. When g is monotonic increasing there is a shortcut, $F_Y(y) = F_X\big(g^{-1}(y)\big)$.

> **Worked example.** With $f_X(x) = \frac{1}{2}x$ on $[0, 2]$ (so $F_X(x) = \frac{1}{4}x^2$), let $Y = X^2$. For $0 \leqslant y \leqslant 4$, $F_Y(y) = F_X(\sqrt{y}) = \frac{1}{4}y$, so $f_Y(y) = F_Y'(y) = \frac{1}{4}$ – that is, Y is uniform on $[0, 4]$.

Inference using normal and t-distributions

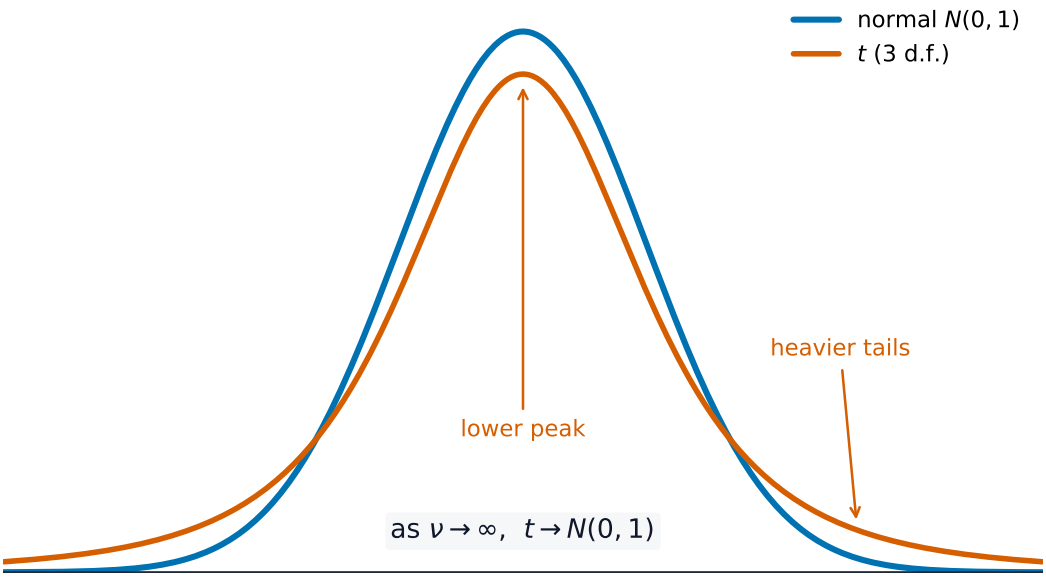


A Galton board: balls falling through pins pile up into the bell-shaped normal distribution.

When a sample is small and the population variance is unknown, base your **hypothesis test** on the t-distribution instead of the normal. The same idea gives a **confidence interval** for the mean:

$$\bar{x} \pm t \frac{s}{\sqrt{n}},$$

where t comes from the t-tables with $n - 1$ degrees of freedom. To compare two populations, use a **two-sample** (2-sample) or **paired-sample** t-test, after finding a **pooled estimate** of the shared variance when appropriate.



For a small sample the t-distribution is flatter with heavier tails, so its critical values are larger.

Worked example. A sample of $n = 10$ has mean $\bar{x} = 50$ and standard deviation $s = 4$. Find a 95% confidence interval for the mean (use $t = 2.262$ for 9 degrees of freedom).

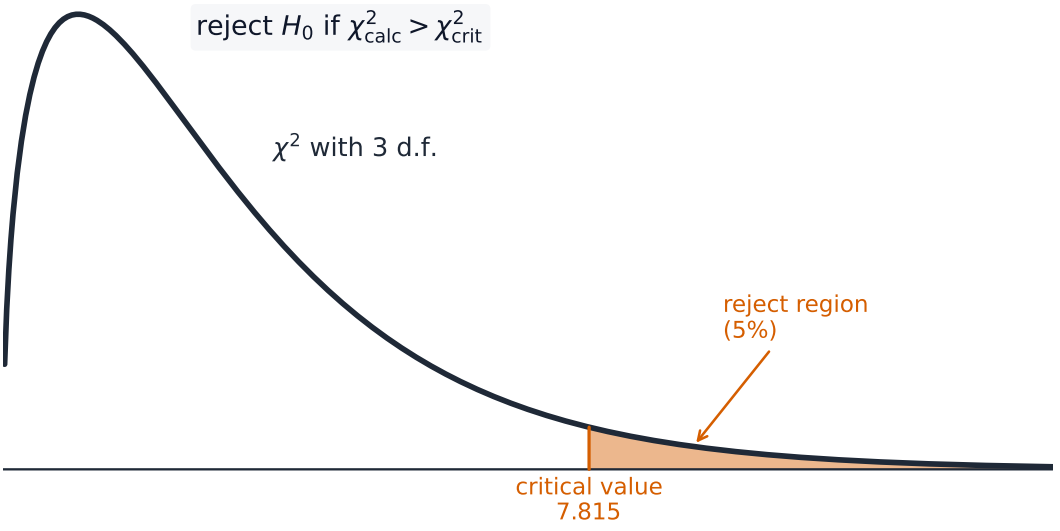
$$50 \pm 2.262 \times \frac{4}{\sqrt{10}} = 50 \pm 2.86 \;\Rightarrow\; (47.1,\ 52.9).$$

Chi-squared tests

A χ^2-test (**chi-squared test**) compares observed counts O with expected counts E from a **theoretical distribution**:

$$\chi^2 = \sum \frac{(O - E)^2}{E}.$$

Compare this with a table value for the right number of **degrees of freedom**. Two uses: a **goodness of fit** test (does the data follow the proposed model?), and a test for **independence** of two variables in a **contingency table**.



The test rejects the model when χ^2 exceeds the critical value, landing in the shaded 5% tail.

Worked example. Four equally likely categories give observed counts $20, 30, 25, 25$ (so each expected count is 25). Test the fit at the 5% level.

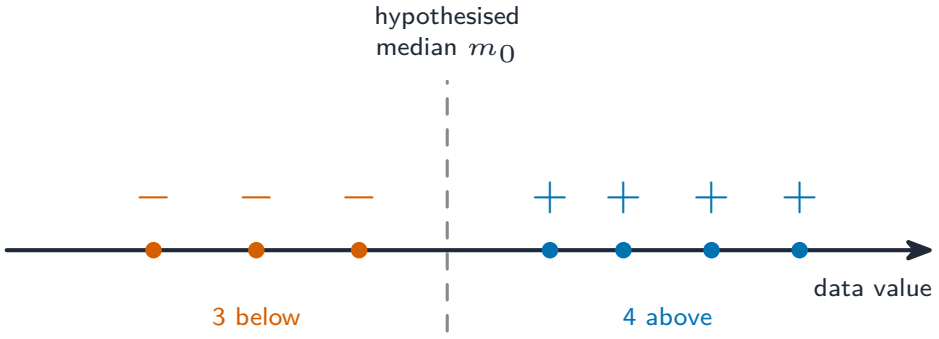
$$\chi^2 = \frac{(20-25)^2 + (30-25)^2 + 0 + 0}{25} = \frac{25+25}{25} = 2.$$

With $4 - 1 = 3$ degrees of freedom the table value is 7.815. Since $2 < 7.815$, do not reject the model.

Non-parametric tests

A **non-parametric test** makes no assumption that the data is normal, so it is useful when that assumption fails. The basic ones are:

- the **sign test**: count how many values fall above and below a proposed median, and test those counts with a binomial model;
- the **Wilcoxon signed-rank test** (the **matched-pairs** test for paired data), which also uses the sizes of the differences, not just their signs;
- the **Wilcoxon rank-sum test**, for comparing two separate samples.



The sign test counts how many values fall above and below the proposed median, then tests those counts with a binomial model

> **Worked example.** Test whether a median is 5. In a sample of 10 values (none equal to 5), 9 lie **above** 5 and 1 lies below. Test at the 5% level (two-tailed).

Under H_0 (median $= 5$) the number above follows $B(10, 0.5)$. The observed result (9 above) is extreme, so find $P(X \geq 9) = \binom{10}{9}(0.5)^{10} + (0.5)^{10} = \dfrac{11}{1024} = 0.0107$. For a two-tailed test compare with $\frac{1}{2}(5\%) = 0.025$. Since $0.0107 < 0.025$, **reject** H_0: there is evidence the median is not 5.

Probability generating functions



Dice: a starting point for probability and discrete random variables.

The **probability generating function** of a discrete variable X is

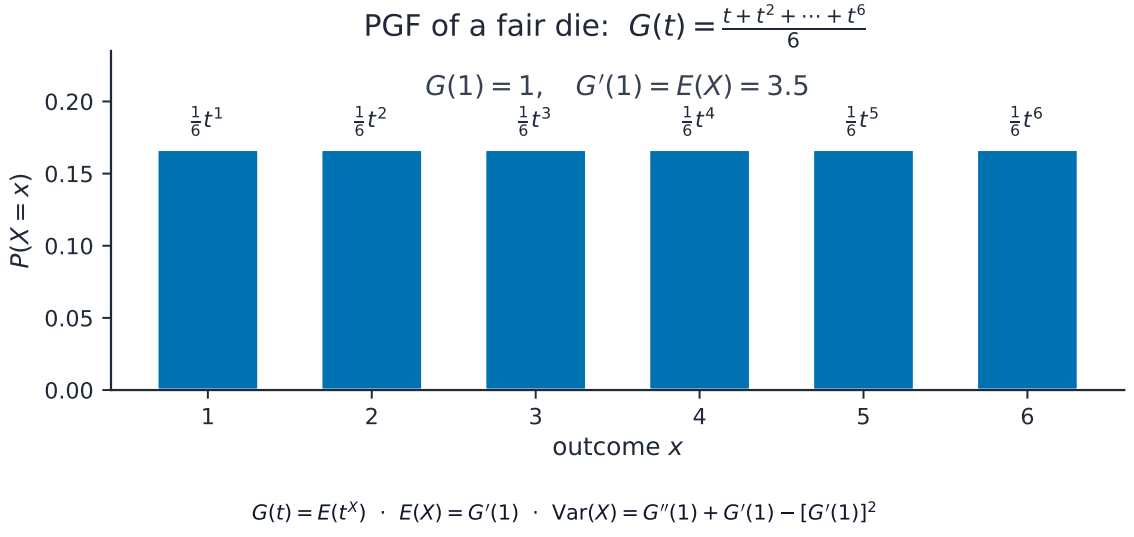
$$G(t) = E(t^X) = \sum_x P(X = x)\, t^x.$$

It packs the whole distribution into one function. The mean and variance come from its derivatives at $t = 1$: $E(X) = G'(1)$ and $\mathrm{Var}(X) = G''(1) + G'(1) - \big(G'(1)\big)^2$. Also, the PGF of a sum of independent variables is the product of their PGFs.

> **Worked example.** X has $P(X = 0) = 0.5$, $P(X = 1) = 0.3$, $P(X = 2) = 0.2$. Find $E(X)$ using the PGF.

Here $G(t) = 0.5 + 0.3t + 0.2t^2$, so $G'(t) = 0.3 + 0.4t$ and

$$E(X) = G'(1) = 0.3 + 0.4 = 0.7.$$



PGF of a fair die packs equal masses at 1 through 6 into $G(t)$

Exam tips

- For a **continuous random variable**, the pdf integrates to 1 over its range, and $E(X) = \int x f(x)\, dx$.
- Use the ***t*-distribution** when the sample is small and the population variance is unknown; state the degrees of freedom.
- For a **chi-squared** test compute $\sum (O - E)^2/E$, compare with the critical value at the right degrees of freedom, and **combine classes with** $E < 5$.
- State H_0 and H_1 and give the conclusion **in context** for every test.