T1 programming fundamentals	T1 type	T1 value
T1 variables	T1 string	T1 selection
T1 iteration	T1 infinite loop	T1 trace table

T1 An amount, number or item assigned to a variable or expression, or held by a data object.	T1 The kind of value a piece of data is, such as integer, string or list, which decides what operations it allows.	T1 The opening unit of this course: variables, control flow, functions and basic data structures.
T1 Choosing instructions by a condition; in relational algebra, choosing rows that satisfy a condition.	T1 A sequence of characters treated as a single data value.	T1 Names bound to values in a program.
T1 A table recording variable values and outputs as an algorithm's steps are followed.	T1 A loop whose stopping condition is never met, so it runs forever.	T1 Repeating a sequence of instructions or calculations.

T1 function	T1 return value	T1 parameter
T1 argument	T1 local scope	T1 list
T1 dictionary	T1 tuple	T1 set

T1 A name in a function definition that receives a supplied argument value.	T1 The result a function sends back to the code that called it.	T1 A callable unit of program code that performs a task and may return a result.
T1 An ordered, changeable collection of values, accessed by position.	T1 The part of a program, usually one function, where a variable created inside it can be used.	T1 Values or references supplied to a function or procedure when it is called.
T1 A collection of distinct elements.	T1 An ordered collection of values that cannot be changed after it is created.	T1 A data structure that stores associations between keys and their values.

T1 aliasing	T1 file	T1 exceptions
T1 decomposition	T1 testing	T1 debugging
T2 discrete mathematics	T2 truth table	T2 predicate logic

T1 Signals raised during execution to report conditions that can be handled by calling code.	T1 A named collection of data stored and managed as a unit.	T1 When two variable names refer to the same object, so a change made through one is seen through the other.
T1 Finding and fixing the cause of an error in a program.	T1 Running a program with chosen inputs to check that it gives the expected outputs.	T1 Breaking a computing problem into smaller tasks that can be solved separately.
T2 Logic using predicates, variables and quantifiers to express statements about objects.	T2 A table listing the output of a logical expression or circuit for every combination of inputs.	T2 The study of distinct structures such as logical statements, sets, graphs and integers.

T2 direct proof	T2 contraposition	T2 contradiction
T2 power set	T2 injective	T2 surjective
T2 bijective	T2 equivalence relation	T2 base case

T2 A proof method that assumes the claim is false and derives an impossibility.	T2 Proving an implication by proving that a false conclusion implies a false premise.	T2 A proof deriving its conclusion directly from assumptions and established results.
T2 Describing a function that reaches every element of its target set.	T2 Describing a function in which distinct inputs have distinct outputs.	T2 The set of all subsets of a given set, including the empty set.
T2 An initial case proved directly in an induction argument, or a simple case that ends a recursive process.	T2 A relation that is reflexive, symmetric and transitive.	T2 Describing a function that is both injective and surjective.

T2 inductive step	T2 induction hypothesis	T2 addition rule
T2 multiplication rule	T2 permutation	T2 combination
T2 inclusion–exclusion	T2 handshake lemma	T2 adjacency matrix

T2 Counting disjoint alternatives by adding their numbers of choices.	T2 The assumption that a claim holds for the current case in an induction proof.	T2 The part of an induction proof deriving the next case from the assumed current case.
T2 A selection of items in which their order is not significant.	T2 An ordered arrangement of selected objects, in which changing the order gives a different arrangement.	T2 Counting successive choices by multiplying their numbers of possibilities when each stage has the stated number.
T2 A square table recording edges between pairs of graph vertices.	T2 In a finite undirected graph, the sum of vertex degrees equals twice the number of edges.	T2 Counting a union by adding set sizes and correcting for overlaps.

T2 adjacency list	T2 tree	T2 BFS
T2 DFS	T2 sample space	T2 conditional probability
T2 Bayes' theorem	T3 two's complement	T3 unsigned

T2 Breadth-first search: visiting a graph level by level, usually with a queue.	T2 A connected undirected graph containing no cycles.	T2 A graph representation storing a list of neighbours for each vertex.
T2 The probability of an event given that another event has occurred.	T2 The set of all possible outcomes of a random experiment.	T2 Depth-first search: following a branch before returning to explore alternatives.
T3 Of an integer type: representing only zero and positive values.	T3 A signed binary representation in which the highest bit has negative positional weight.	T2 A rule relating a conditional probability to its reverse conditional probability and prior probabilities.

T3 IEEE-754	T3 ASCII	T3 Unicode
T3 NAND	T3 half adder	T3 full adder
T3 multiplexer	T3 fetch-decode-execute cycle	T3 cache

T3 The standard giving every character in every writing system a unique number.	T3 A character code with 128 code values represented in seven bits.	T3 A standard specifying floating-point representations, arithmetic and exceptional values.
T3 A circuit that adds two bits and a carry-in, giving a sum bit and a carry-out.	T3 A logic circuit that adds two single binary digits and produces a sum bit and a carry bit.	T3 A logic operation whose output is false only when all its inputs are true.
T3 Fast storage holding copies of frequently needed data or instructions to reduce access time.	T3 The repeated processor steps of obtaining an instruction, interpreting it and carrying it out.	T3 A circuit that chooses one of several inputs and passes it to its output.

T3 locality	T3 addressing mode	T3 process
T3 context switch	T3 virtual memory	T3 page fault
T3 system call	T3 linker	T3 assembler

T3 An instance of a program in execution together with its execution state and allocated resources.	T3 The rule an instruction uses to specify where an operand is or how its address is obtained.	T3 The tendency to reuse recently accessed locations or nearby locations.
T3 An exception raised when a memory access cannot be completed using the current page mapping, for example because the required page is not in physical memory.	T3 A memory-management system that maps virtual addresses to physical storage and may move pages between RAM and secondary storage.	T3 Saving one task's execution state and restoring another's so the processor can change which task it runs.
T3 Software that translates assembly-language instructions into machine code.	T3 A tool that combines object code and resolves references between compiled modules or libraries.	T3 An interface through which a program requests a service from the operating-system kernel.

T3 preprocessor	T3 computer systems fundamentals	T3 overflow
T3 rounding	T3 compiler	T4 object-oriented programming
T4 class	T4 object	T4 instance

T3 A condition in which a result exceeds the range representable by the available storage or number format.	T3 How data is represented and how hardware, the operating system and programs work together.	T3 A tool that processes source directives, such as C macros and conditional inclusion, before compilation.
T4 Organising programs around objects that combine state and behaviour.	T3 Software that translates a program from a source language into another form, commonly machine or intermediate code.	T3 Replacing a value with the nearest representable one, a source of floating-point error.
T4 A particular occurrence or example; in object-oriented programming, an object created from a class.	T4 An entity in a program with associated state and behaviour, often an instance of a class.	T4 A definition describing the state and behaviour of its instances.

T4 instance attributes	T4 Class attributes	T4 reference
T4 aliases	T4 equality	T4 identity
T4 invariant	T4 encapsulation	T4 inheritance

T4 A value identifying an object; assigning it to another Python name does not copy the object.	T4 Attributes stored on a class, which instances may access through attribute lookup.	T4 Values associated with individual objects rather than stored on their class.
T4 Whether two references identify the very same object.	T4 A comparison of values according to an object's equality operation.	T4 Different names or references that identify the same object.
T4 A class relationship in which a subclass derives behaviour and attributes from a base class.	T4 Grouping data with the operations that use it and controlling access through an interface.	T4 A property required to remain true at specified points, such as before and after public method calls.

T4 composition	T4 fragile base class	T4 interface
T4 polymorphism	T4 duck typing	T4 dynamic dispatch
T4 substitution	T4 exceptions	T4 unit tests

T4 The operations and contracts through which other code uses an object or component.	T4 A base class whose changes can unexpectedly break subclasses that depend on its implementation.	T4 Building an object by combining it with other objects that provide parts of its behaviour or state.
T4 Selecting a method implementation at runtime according to the object's type.	T4 Accepting an object based on the operations it supports rather than a declared class relationship.	T4 Using a common operation with different implementations selected for different object types.
T4 Tests of individual units of behaviour, usually isolated from unrelated parts of the system.	T4 Signals raised during execution to report conditions that can be handled by calling code.	T4 Using an implementation or subtype in place of another while preserving the required contract.

T4 coupling	T4 cohesion	T4 single responsibility principle
T4 runtime type	T4 open/closed	T4 arrange
T4 act	T4 assert	T5 data structures and algorithms

T4 The design principle that a class should have one coherent responsibility and reason to change.	T4 How closely related the responsibilities within a software component are.	T4 The degree of dependency between software components.
T4 The first step of a unit test: set up the objects and data.	T4 The principle that code should be open for extension but closed for modification.	T4 The actual class of an object while the program runs, which decides which method is called.
T5 Ways of organising data and the procedures used to process it.	T4 The third step of a unit test: check that the result is what was expected.	T4 The second step of a unit test: call the behaviour being tested.

T5 Big-O	T5 Big-Θ	T5 array
T5 linked list	T5 node	T5 stack
T5 queue	T5 hash function	T5 collision

T5 An indexed collection of elements arranged in one or more dimensions.	T5 An asymptotically tight bound giving both upper and lower growth bounds up to constant factors.	T5 An asymptotic upper bound on growth, ignoring constant factors for sufficiently large inputs.
T5 A last-in, first-out data structure in which items are added and removed at the same end.	T5 An individual unit or point in a network, graph, tree or linked data structure.	T5 A sequence of nodes connected by links rather than located by consecutive array positions.
T5 A conflict such as different keys mapping to the same hash location, or simultaneous transmissions interfering on a shared network medium.	T5 A function mapping a key to a hash value used to choose or locate a table position.	T5 A data structure in which items are normally removed in the order they were added.

T5

chaining

T5

open addressing

T5

load factor

T5

clustering

T5

binary search tree

T5

heap

T5

min-heap

T5

adjacency list

T5

adjacency matrix

T5 The number of stored hash-table entries divided by the number of buckets or slots.	T5 Handling hash collisions by probing other positions within the table.	T5 Handling hash collisions by storing multiple entries at one bucket.
T5 A tree-based structure satisfying an ordering rule between each parent and its children.	T5 A binary tree ordered so keys in the left subtree precede the node and keys in the right follow it.	T5 Groups of occupied hash-table positions that can lengthen probe sequences.
T5 A square table recording edges between pairs of graph vertices.	T5 A graph representation storing a list of neighbours for each vertex.	T5 A heap in which every parent is no greater than its children, placing a minimum at the root.

T5 inversion	T5 insertion sort	T5 merge sort
T5 linear search	T5 binary search	T6 computer architecture
T6 instruction set architecture	T6 RISC	T6 load-store

T5 Sorting by recursively sorting parts and merging the sorted parts.	T5 A sorting method that inserts each new item into its correct position in the already sorted part of a list.	T5 A pair of sequence positions whose values occur in the opposite order to the desired sorted order.
T6 The organisation and programmer-visible behaviour of computer hardware.	T5 A search of ordered data that repeatedly compares with the middle item and discards half of the remaining search range.	T5 Checking items in sequence until the required item is found or the list ends.
T6 An instruction-set organisation in which memory data is accessed through load and store instructions.	T6 Reduced instruction set computer: an instruction-set design approach emphasising comparatively simple operations.	T6 The programmer-visible instructions, registers, data formats and rules a processor implements.

T6

byte-addressable

T6

datapath

T6

control

T6

ALU

T6

pipelining

T6

retires

T6

CPI

T6

read-after-write hazard

T6

forwarding

T6 The processor logic that directs datapath operations using control signals.	T6 Processor components and connections that store, carry and operate on instruction data.	T6 Having a distinct memory address for each byte.
T6 Commits an instruction's result to the processor state after successful execution.	T6 Overlapping different stages of several instructions to improve throughput.	T6 The part of a processor that performs arithmetic and logical operations.
T6 Sending a result directly to a dependent pipeline stage before normal register write-back.	T6 A dependency in which an instruction needs a value an earlier instruction has not yet made available.	T6 Cycles per instruction: processor cycles divided by the number of executed instructions.

T6 memory hierarchy	T6 locality	T6 direct-mapped cache
T6 tag	T6 index	T6 block offset
T6 average memory access time	T6 evicts	T6 programmed I/O

T6 A cache in which each memory block has exactly one possible cache location.	T6 The tendency to reuse recently accessed locations or nearby locations.	T6 Levels of storage with different speed, capacity and cost characteristics.
T6 The address bits selecting a byte or position within a memory block.	T6 The address bits selecting a cache set or line in this architecture topic.	T6 The address information used to identify which memory block occupies a cache location.
T6 Input/output in which processor instructions transfer data and check device status.	T6 Removes a cache block to make space for another block.	T6 The expected memory-access cost, combining hit time with miss probability and miss penalty.

T6

interrupt-driven I/O

T6

DMA

T6

instruction-level parallelism

T6

SMT

T6

R-type

T6

I-type

T7

operating systems

T7

process

T7

states

T6 Overlapping or executing independent instructions within one instruction stream.	T6 Direct memory access: hardware transfers data between memory and a device without processor instructions for each item.	T6 Input/output in which device interrupts notify the processor that handling is needed.
T6 A MIPS instruction format with an immediate (constant) operand.	T6 A MIPS instruction format with three register operands.	T6 Simultaneous multithreading: instructions from multiple hardware threads share one processor core.
T7 The possible configurations or conditions of a system at a particular point in its operation.	T7 An instance of a program in execution together with its execution state and allocated resources.	T7 Software that manages hardware resources and provides services to running programs.

T7

ready

T7

running

T7

blocked

T7

process control block

T7

context switch

T7

thread

T7

FCFS

T7

SJF

T7

round robin

T7 A task state waiting for an event or resource before it can become ready.	T7 A task state in which its instructions are currently executing on a processor.	T7 A scheduling state in which a task can run once it receives processor time.
T7 An execution flow with its own execution state that can share a process's resources.	T7 Saving one task's execution state and restoring another's so the processor can change which task it runs.	T7 An operating-system record containing the state and management information associated with a process.
T7 Scheduling ready tasks in turns with a limited time quantum.	T7 Shortest-job-first scheduling: choosing the ready task with the shortest predicted CPU burst.	T7 First-come, first-served scheduling: choosing ready tasks in arrival order.

T7

Gantt chart

T7

turnaround time

T7

waiting time

T7

interleaving

T7

mutual exclusion

T7

lock

T7

spin

T7

virtual address

T7

page table

T7 Time a job spends in the ready queue waiting for processor service.	T7 Elapsed time from a job's arrival to its completion.	T7 A timeline showing when scheduled tasks use the processor.
T7 A synchronisation mechanism used by participating tasks to control access to shared resources.	T7 Ensuring that at most one participating task enters a protected critical section at a time.	T7 An ordering of steps from multiple tasks that preserves each task's required internal order.
T7 A structure recording virtual-page mappings and access information.	T7 An address in a process's virtual address space, translated or checked by memory-management hardware.	T7 Repeatedly checking for a condition while continuing to consume processor time.

T7

physical frame

T7

TLB

T7

page fault

T7

inode

T7

journaling

T7

rings

T7

kernel mode

T7

system call

T7

least privilege

T7 An exception raised when a memory access cannot be completed using the current page mapping, for example because the required page is not in physical memory.	T7 Translation lookaside buffer: a cache of recent virtual-to-physical address translations.	T7 A fixed-size block of physical memory that can hold a page.
T7 Hardware privilege levels controlling access to protected processor operations.	T7 Recording filesystem changes in a log to support recovery after interruption.	T7 A filesystem record storing file metadata and information locating file data.
T7 Granting only the permissions needed to carry out an authorised task.	T7 An interface through which a program requests a service from the operating-system kernel.	T7 A privileged processor mode used to run trusted operating-system code.

T7 FAT	T8 databases	T8 entity
T8 relationship	T8 cardinality	T8 primary key
T8 foreign key	T8 referential integrity	T8 selection

T8 An identifiable thing represented in a data model, such as a student or course.	T8 An organised collection of data that can be accessed and managed.	T7 File Allocation Table: a table that links a file's disk blocks in a chain.
T8 A field or combination of fields that uniquely identifies each record in a database table.	T8 The number of elements in a collection; in database relationships, the permitted numbers of records associated with a record of another entity.	T8 An association between entities in a data model.
T8 Choosing instructions by a condition; in relational algebra, choosing rows that satisfy a condition.	T8 Ensuring that a non-null foreign-key reference identifies an existing referenced key.	T8 A field or group of fields that refers to a candidate key in another or the same database table.

T8 projection	T8 join	T8 insert anomaly
T8 update anomaly	T8 delete anomaly	T8 functional dependency
T8 second normal form	T8 third normal form	T8 Boyce-Codd

T8 A design problem in which adding one fact requires adding unrelated or unavailable facts.	T8 An SQL operation that combines rows from tables according to a specified relationship or condition.	T8 A relational-algebra operation choosing specified columns from a relation.
T8 A constraint under which each value of one attribute set determines one value of another attribute set.	T8 A design problem in which removing one fact unintentionally removes another needed fact.	T8 A design problem in which one fact is repeated and must be changed consistently in several rows.
T8 A normal form requiring the determinant of every nontrivial functional dependency to be a superkey.	T8 A relation where every nontrivial functional dependency has a superkey determinant or a prime right-hand attribute.	T8 First normal form with no non-prime attribute depending on only part of any candidate key.

T8 B-tree	T8 query plan	T8 transaction
T8 atomicity	T8 consistency	T8 isolation
T8 durability	T8 serializability	T8 isolation levels

T8 A group of database operations treated as one unit with defined commit or rollback behaviour.	T8 The sequence of operations a database chooses to execute a query.	T8 A balanced multiway search tree storing ordered keys and supporting efficient lookup and update.
T8 The degree to which concurrent transactions are protected from one another's intermediate effects.	T8 The requirement that a correctly defined transaction preserves database constraints.	T8 The transaction property that its changes take effect together or are rolled back together.
T8 Specified concurrency guarantees defining which transaction anomalies are prevented or allowed.	T8 Equivalence of a concurrent transaction schedule to some serial schedule under the chosen equivalence criterion.	T8 The guarantee that committed changes survive the failures covered by the database's recovery model.

T8

two-phase locking

T8

authentication

T8

authorisation

T8

principle of least privilege

T8

parameterised statements

T8

data governance

T9

computer networks

T9

layering

T9

service

T8 The granting or checking of permission to access a resource or perform an action.	T8 Checking the identity claimed by a user, device or process.	T8 A locking protocol that permits no new lock acquisition after the first lock release.
T8 Rules, responsibilities and processes for managing data quality, access, use and retention.	T8 SQL statements that bind supplied data values separately from the query's syntax.	T8 Giving each user or process only the access needed to carry out its authorised tasks.
T9 The communication capability one protocol layer offers to the layer above it.	T9 Organising communication functions into levels that use services from lower levels.	T9 Connected devices and communication links that exchange data using agreed protocols.

T9 encapsulated	T9 physical	T9 link
T9 network	T9 transport	T9 application
T9 MAC address	T9 CRC	T9 TTL

T9 The network layer concerned with transferring frames across a local link.	T9 The network layer concerned with transmitting bits as signals over a medium.	T9 Packaged within the protocol information needed by a lower communication layer.
T9 The layer containing protocols used by applications, such as HTTP and DNS.	T9 The protocol layer providing communication between application processes, such as through TCP or UDP.	T9 A set of connected devices or nodes that can exchange information.
T9 The IP hop-limit field that routers reduce to stop packets circulating indefinitely.	T9 Cyclic redundancy check: an error-detection value calculated from transmitted bits.	T9 A link-layer address associated with a network interface.

T9 IP address	T9 routing table	T9 longest prefix match
T9 UDP	T9 TCP	T9 three-way handshake
T9 DNS	T9 HTTP	T9 status codes

T9 Choosing the routing-table entry whose matching destination prefix is longest.	T9 Stored information a router uses to choose where to forward packets for different network destinations.	T9 An address identifying an interface within an IP network.
T9 The SYN, SYN-ACK and ACK exchange normally used to establish a TCP connection.	T9 A transport protocol providing a reliable ordered byte stream with flow and congestion control.	T9 A transport protocol providing datagrams and ports without guaranteeing delivery or order.
T9 Numbers in HTTP responses describing the outcome of a request.	T9 An application protocol using requests and responses to transfer representations of resources.	T9 The distributed domain name system used to look up records such as addresses for names.

T9 symmetric	T9 asymmetric	T9 TLS
T9 MAC	T9 distributed system	T9 partial failure
T9 idempotence	T9 happens-before	T9 Lamport clock

T9 A protocol that protects data in transit using encryption, integrity checks and peer authentication.	T9 In cryptography, describing methods that use a related public key and private key rather than one shared secret key.	T9 Describing encryption in which communicating parties share a secret key for encryption and decryption.
T9 Failure of some components while other components continue operating.	T9 Independent computers coordinating through messages to provide a service or perform work.	T9 Message authentication code: a keyed value used to check message integrity and authenticity.
T9 A logical counter scheme assigning timestamps consistent with the happens-before relation.	T9 A causal ordering induced by local event order and message transmission, including transitive consequences.	T9 The property that repeating an operation has the same effect as performing it once.

T9 confidentiality	T10 algorithm design and analysis	T10 recurrence
T10 recursion tree	T10 merge sort	T10 quicksort
T10 greedy algorithm	T10 exchange argument	T10 dynamic programming

T10 An equation expressing a quantity, such as running time, using values for smaller inputs.	T10 Constructing algorithms and studying their correctness and resource costs.	T9 Keeping information accessible only to those authorised to see it.
T10 Sorting by partitioning values around a pivot and recursively sorting the partitions.	T10 Sorting by recursively sorting parts and merging the sorted parts.	T10 A diagram of recursive calls used to add work across levels of a recurrence.
T10 Solving overlapping subproblems and storing their results for reuse.	T10 A proof showing that choices in an optimal solution can be replaced by proposed choices without worsening it.	T10 An algorithm making a locally preferred choice at each step without revisiting earlier choices.

T10 DP table	T10 dependency order	T10 Kruskal
T10 cuts	T10 reduction	T10 NP-completeness
T11 theory of computation and compilers	T11 deterministic finite automaton	T11 pumping lemma

T10 A minimum-spanning-tree algorithm adding edges in increasing weight order while avoiding cycles.	T10 An order in which every needed subproblem result is available before it is used.	T10 A table storing solutions to subproblems in a dynamic-programming algorithm.
T10 The property of being in NP and at least as hard as every NP problem under polynomial-time reductions.	T10 A transformation relating the solution of one problem to the solution of another.	T10 Partitions of a graph's vertex set into two groups.
T11 A necessary property of regular languages used to prove that some languages are not regular.	T11 A finite-state machine with one transition for each state and input symbol.	T11 The study of formal computing models, languages and the translation of programs.

T11 context-free grammars	T11 derivation	T11 parse tree
T11 ambiguity	T11 Turing machine	T11 halting problem
T11 diagonalisation	T11 tokens	T11 context-free languages

T11 A tree recording how grammar productions generate the structure of a string.	T11 A sequence of production-rule applications that generates a string from a grammar.	T11 Grammars whose production rules replace a single nonterminal with a sequence of terminals and nonterminals.
T11 The undecidable problem of determining for every program and input whether execution eventually stops.	T11 An abstract computing machine with finite control and a read-write tape.	T11 The existence of more than one parse tree for a string under a given grammar.
T11 Languages generated by context-free grammars; these include all regular languages and some nonregular languages.	T11 Lexical units such as identifiers, keywords, literals and operators produced by a lexer.	T11 A construction differing from each member of a proposed list at a selected position.

T11 symbol table	T11 intermediate representation	T12 software engineering
T12 requirements engineering	T12 functional requirement	T12 non-functional requirement
T12 use case	T12 layers	T12 dependency rule

T12 The systematic design, development, testing and maintenance of software.	T11 An internal program representation connecting stages of a compiler.	T11 A compiler data structure recording identifiers and information such as their types, scopes or locations.
T12 A constraint or quality requirement, such as response time, availability or security.	T12 A statement of a behaviour or service the system must provide.	T12 Finding, analysing, agreeing and managing what a software system must provide.
T12 In clean architecture, the rule that source-code dependencies point toward inner policies rather than outer mechanisms.	T12 Software levels separating concerns such as presentation, domain logic and storage.	T12 A description of interactions by which an actor achieves a goal using a system.

T12 cohesion	T12 coupling	T12 states
T12 transitions	T12 branching model	T12 merge conflicts
T12 test pyramid	T12 user stories	T12 standup

T12 The possible configurations or conditions of a system at a particular point in its operation.	T12 The degree of dependency between software components.	T12 How closely related the responsibilities within a software component are.
T12 Overlapping edits that version control cannot combine automatically and requires a person to resolve.	T12 A version-control workflow defining how branches are created, used and integrated.	T12 Changes from one state to another in response to events or actions.
T12 A short team meeting to coordinate progress, next steps and blockers.	T12 Short descriptions of user goals and value, supported by acceptance criteria.	T12 A testing guideline favouring many fast unit tests and fewer broader integration and end-to-end tests.

T12 retro	T12 threat model	T13 artificial intelligence and machine learning
T13 search tree	T13 uninformed	T13 breadth-first
T13 depth-first	T13 informed	T13 heuristic

T13 Methods for building systems that reason, search, predict or learn patterns from data.	T12 A structured description of assets, possible threats and controls addressing them.	T12 A retrospective meeting in which a team reviews its process and chooses improvements.
T13 Exploring all nodes at one depth before proceeding to the next depth.	T13 Describing search that uses no problem-specific estimate of the distance or cost to a goal.	T13 A tree of states and possible moves explored when solving a search problem.
T13 An estimate used to guide search, such as an estimate of the remaining path cost.	T13 Describing search guided by problem-specific estimates of remaining cost or distance.	T13 Exploring a branch as far as required before returning to alternatives.

T13 symbolic reasoning	T13 propositional logic	T13 inference
T13 forward chaining	T13 backward chaining	T13 Bayesian network
T13 linear regression	T13 logistic regression	T13 decision boundary

T13 Deriving conclusions from premises using a reasoning rule or model.	T13 Logic combining statements with truth values using logical connectives.	T13 Deriving conclusions by manipulating explicit symbols, facts and rules.
T13 A directed acyclic graph representing a joint probability distribution through conditional probabilities.	T13 Starting with a goal and working backward to facts or subgoals that could support it.	T13 Applying rules to known facts to derive additional conclusions.
T13 The boundary between input regions assigned different predicted classes.	T13 A model mapping a linear score through a logistic function to estimate a binary-class probability.	T13 A method of modelling a response using a linear combination of explanatory variables and fitted coefficients.

T13 gradient descent	T13 learning rate	T13 loss surface
T13 confusion matrix	T13 accuracy	T13 precision
T13 recall	T13 error disparity	T13 A*

T13 The loss value viewed as a function of a model's parameter values.	T13 The step-size factor controlling the scale of a parameter update during training.	T13 An iterative method that adjusts parameters in the direction opposite the gradient to seek a smaller value of an objective function.
T13 True positives divided by all predicted positives, when that denominator is nonzero.	T13 The number of correct predictions divided by the total number of predictions.	T13 A table counting predictions for each combination of actual and predicted class.
T13 A best-first search that expands the node with the lowest path cost plus heuristic estimate.	T13 A difference in model error rates between groups.	T13 True positives divided by all actual positives, when that denominator is nonzero.