Computational thinking

IB DP Computer Science B1 Computational thinking full package

How do you plan a whole app?

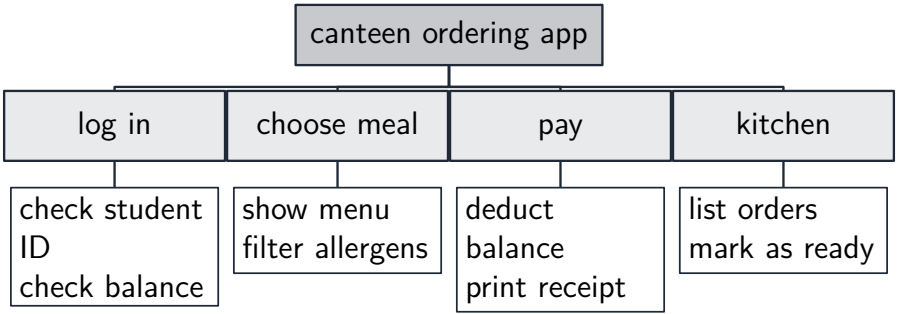
> Big and vague becomes small and exact.

- Theme B1: computational thinking, for SL and HL.
- Theme B2 then turns these ideas into Python.

Four approaches

Approach	**Canteen example**
decomposition	log in, choose meal, pay, kitchen
pattern recognition	every screen shows a list and takes a choice
abstraction	a meal is a name, a price and allergens
algorithmic design	exact steps to deduct a price

Decomposing the canteen app



> "Check balance" serves both logging in and paying.

Patterns and abstraction

Pattern recognition

Find the same structure in different places.

Lunch, kit and trip payments: add prices, deduct from a balance.

Abstraction

Build a **model** that leaves out what does not matter.

Say what is kept, what is removed, and why.

What makes an algorithm

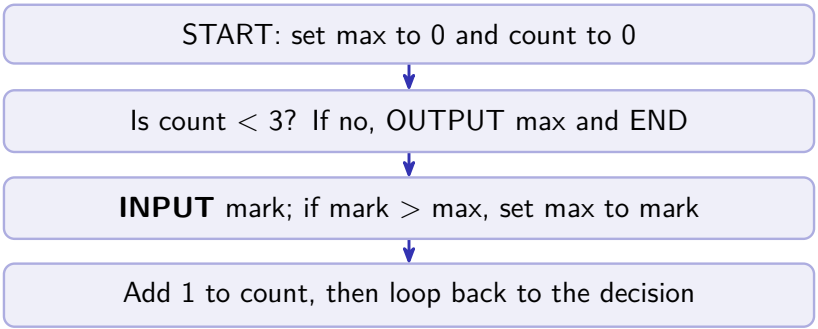
> Exact, finite, correct - with clear inputs and outputs.

- An algorithm is a finite **sequence** of exact steps that solves a problem.
- "Add some salt" is not a step; "add 2 g of salt" is.

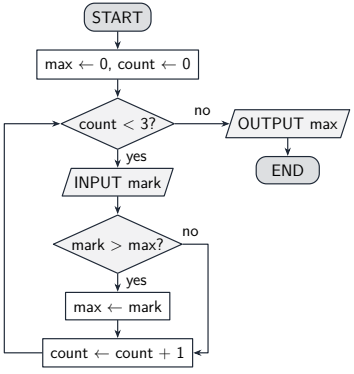
Three building blocks

Construct	Meaning	In a flowchart
sequence	one step after another	arrows joining boxes
selection	a decision picks the path	a diamond: yes or no
iteration	steps repeat	an arrow looping back

Find the largest of three marks



Tracing 62, 85, 71



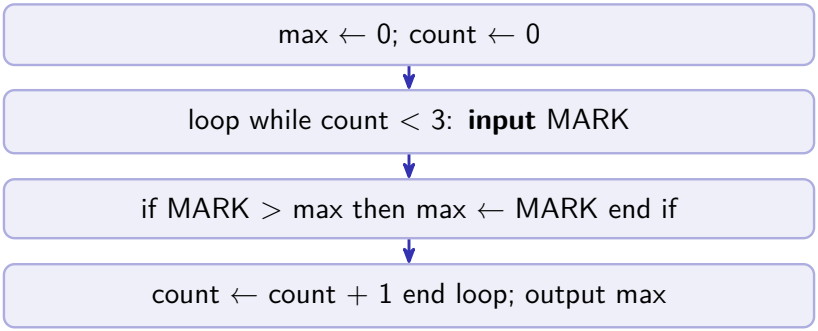
The algorithm outputs 85.

The trap: all negative marks

> Start max at the first mark, not at 0.

- Inputs -5, -8, -2: no mark is greater than 0.
- Check unusual inputs: all equal, all negative, empty, the largest.

The same algorithm as pseudocode



Choosing test data

Type	Purpose	Mark 0 to 100
normal	typical valid data	45, 72
boundary	the edges, and just outside	0, 100, -1, 101
invalid	must be rejected	-20, 150, "ten"

When a test fails

> Is the algorithm, the code or the requirement wrong?

- State the inputs, trace, test the boundaries, compare with the expected output.
- Then decide where the fault lies.

Efficiency

> 1000 checks against about 10.

- Two correct algorithms can differ greatly in speed.
- Theme B2 measures this with Big O.

Practice

Your turn

Try these:

A. Decompose "organise a school sports day".
B. Explain abstraction with a metro map.
C. Trace the flowchart for 40, 40, 90.
D. Test values for a ticket: 12 to age 17, 20 from 18.

Check yourself

