B1 COMPUTATIONAL THINKING · B1.1

Approaches to computational thinking

The mental toolkit before coding: specifying a problem, the four **pillars**, and reading and **tracing flowcharts**.

01 The four pillars

Decomposition	Break into smaller subproblems.
Patterns	Spot similarities and repetition.
Abstraction	Ignore irrelevant detail.
Algorithmic	Design clear, ordered steps.

02 Problem specification

Goal	State exactly what to solve.
Bad	Vague ("our service is inefficient").
Good	Focused, measurable, tied to impact.
Why	Lets you design and test a solution.

03 Flowchart symbols

Terminator	Rounded shape. Marks the start or the end of the algorithm.	START/END
Process	Rectangle. An action or calculation, such as total = total + 1.	ACTION
Decision	Diamond. A yes/no question that branches the flow.	BRANCH
Input / Output	Parallelogram. Reading input or displaying output.	IN/OUT

04 Applying the pillars

Abstraction	Pick out the essentials.
Decompose	Split into subproblems.
Patterns	Use past data and repetition.
Algorithm	Heuristic (fast) or optimal (best).

05 Tracing a flowchart

Trace table	Record each variable, step by step.
Loop check	Condition is tested before each pass.
Off-by-one	Watch the boundary value carefully.
Example	Sum 1 to 3 gives a total of 6.

06 Know the difference

Decomposition vs abstraction	Breaking a problem into parts versus removing irrelevant detail.	PILLARS
Process vs decision	A rectangle (an action) versus a diamond (a yes/no branch).	SHAPES
Heuristic vs optimal	A quick "good enough" answer versus the slower mathematically best one.	ALGORITHM
Specification vs solution	Defining the problem clearly first versus the steps that solve it.	PROCESS

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Eight things that lose marks in Paper 1 if you slip on them. Skim before you walk in.

01

The four pillars: **decomposition, pattern recognition, abstraction, algorithmic thinking**.

02

Decomposition breaks into parts; **abstraction** removes irrelevant detail. Don't swap them.

03

A good problem spec is **focused, measurable, and tied to impact**.

04

Diamond = decision, **rectangle** = process, rounded = start/end, parallelogram = input/output.

05

Trace with a **trace table**: one row per step or loop pass.

06

A loop's condition is checked **before** each pass; watch for off-by-one.

07

A **heuristic** is fast and "good enough"; an **optimal** method is slower but best.

08

Computational thinking turns a messy problem into clear, ordered steps.