2.1 COMPUTATIONAL THINKING · 2.1.1

Thinking abstractly

Removing irrelevant detail to make a problem solvable. Spec 2.1.1(a)–(d).

01 What it is

Def	Representing a problem by **removing/ hiding irrelevant detail** to focus on what matters.
Example	Transit map: keeps stations/lines/order, drops distance & geography.
≠	Not decomposition (= break into sub-problems).

02 Why we need it

Complexity	Reduces it → problem becomes **solvable**.
Focus	On key elements; ignore distractions.
Model	Can be analysed, tested, reused.
General	Same solution fits many similar problems.

03 Abstraction vs reality

Omits	Loses detail/information.
Misleads	If you forget what was removed.
One purpose	Fits its intended use only.
Assumptions	Only as good as these.

04 Devise a model — graph example

Keep	Junctions = nodes; roads = weighted edges.
Drop	Surface, scenery, exact shape.
Result	Routing algorithm works on **any** network.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on abstraction questions.

01

Define it precisely: **remove/hide irrelevant detail**, not just "simplify".

02

State the benefit: **reduces complexity → solvable**.

03

Have a difference from reality ready: it **loses information** & can mislead.

04

In a model, **names/cosmetics = irrelevant**; structure/ constraints = relevant.

05

Don't confuse **abstraction** (remove detail) with **decomposition** (break up).

06

Good go-to example: a **graph** abstracting a road network.