

2.1 COMPUTATIONAL THINKING · 2.1.2

Thinking ahead

Inputs/outputs, preconditions, caching & reusable components. Spec 2.1.2(a)–(d).

01 Inputs & outputs

Inputs Data the system **receives**.

Outputs Results it **produces**.

ATM In: card, PIN, amount · Out: cash, balance, receipt.

02 Preconditions

Def Must be **true before** a solution runs.

Binary search List is **sorted**.

Divide Divisor **≠ 0**.

03 Caching

Def Store recent results/data for **fast reuse**.

How Check cache first: **hit** = instant; **miss** = fetch then store.

+ Faster · less load on CPU/network/DB.

– Extra memory · data can go **stale**.

04 Reusable components

What Subroutine / module / library used again.

Time Write & test once → reuse.

Reliable Already tested = fewer bugs.

Maintain Fix once, everyone benefits.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on "thinking ahead" questions.

01

Inputs come **in**, outputs go **out** — don't swap them.

02

A precondition is a **specific** condition (list sorted), not "data must be correct".

03

"Evaluate caching" needs **drawbacks too**: memory & **stale** data.

04

A cache holds **copies for fast reuse**; checked first, can be cleared.

05

Reuse benefits + **why**: tested → reliable; once → saves time.

06

Examples of components: **sort()**, date module, login module.