

1.2 SOFTWARE & SOFTWARE DEVELOPMENT · 1.2.4

Programming paradigms & procedural languages

Why paradigms exist, their characteristics, and the **procedural** style. Spec 1.2.4(a)(b).

01 The paradigms

Procedural Step-by-step instructions in procedures/functions. Says **how**.

Object-oriented Objects combine data + behaviour.

Declarative Describe **what** you want (SQL, Haskell).

Low-level Maps ~1:1 to machine code (assembly).

02 Why several exist

Fit Different problems suit different styles.

Abstraction Ease of writing vs control over hardware.

Note A paradigm is a **style**, not a language; one language can support several.

03 Procedural characteristics

Structure Sequence · selection (if) · iteration (loops) — the structured building blocks.

Subroutines Decomposed into procedures & functions; reusable, testable in isolation.

Data Kept separate from procedures; passed in as parameters.

Best for Clear step-by-step tasks: calculations, file processing, batch jobs.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on paradigm questions.

01

A paradigm is a **style/approach**, not a language. Python supports several paradigms.

02

Why paradigms exist: different problems suit different styles + the **abstraction trade-off**.

03

Procedural = **how**, step by step. Declarative = **what**, e.g. SQL.

04

Procedural is **structured**: sequence, selection, iteration, subroutines — never "unstructured".

05

"Characteristics" questions want **features** (subroutines, parameters), not just language names.

06

Be ready to **write a short procedural pseudocode program** — it's been asked directly.