

12.1 Program Development Life cycle

Name: _____ Class: _____ Date: _____

Total: 16 marks

Objective

Build the skills to answer exam questions on the **program development life cycle** 开发生命周期—its stages and the common models.

You must be able to:

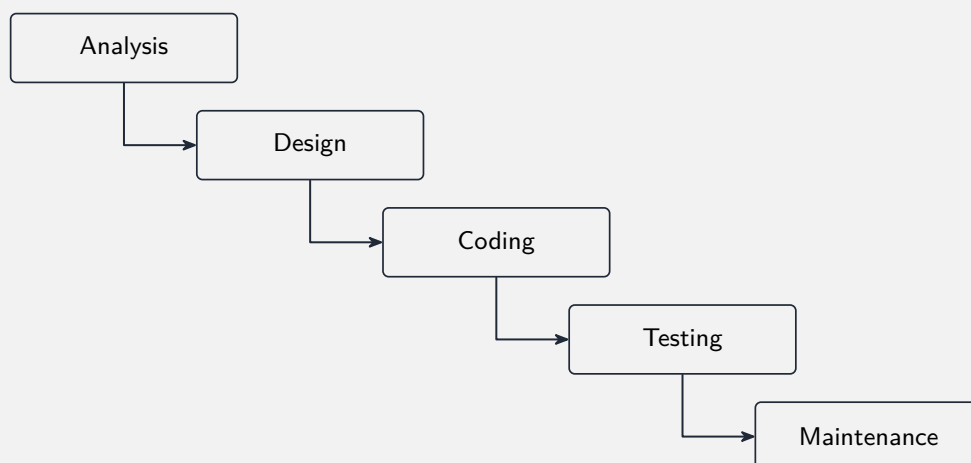
- explain the **purpose** of a development life cycle
- describe the **analysis, design, coding, testing, maintenance** stages
- compare models (**waterfall, iterative, RAD, agile**) and choose one

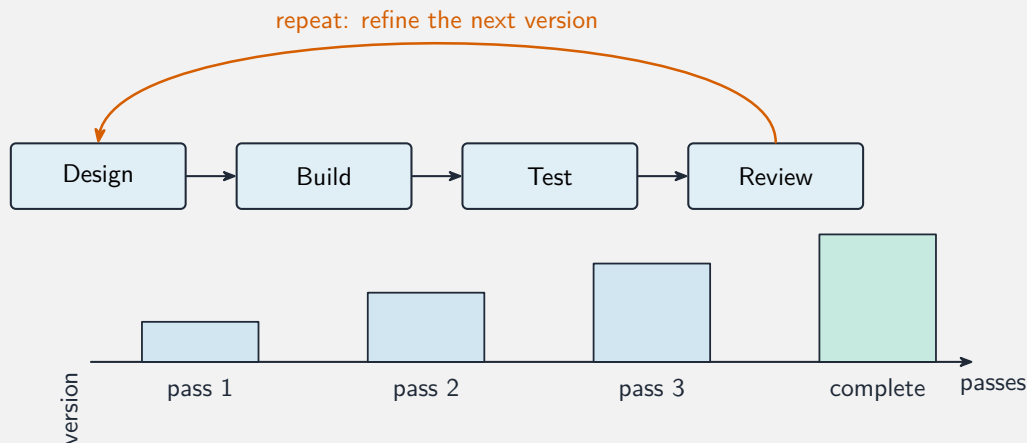
1 Worked examples

Study these first. Each one shows the method for a question type used later —follow the steps and you can do the Practice and Exam-style questions yourself.

■ Purpose and the five stages

A development life cycle is the set of stages from idea to maintained software. It exists to **plan, manage and control** a project —to build the right product, on time, with good quality.





The iterative model contrasts with the waterfall

- **analysis** —find **what** the program must do (gather requirements).
- **design** —decide **how**: data structures, algorithms, modules, interface.
- **coding** —write the source code from the design.
- **testing** —run against test data and fix bugs.
- **maintenance** —after release, keep it working and useful.

Three kinds of **maintenance**: **corrective** 纠正性 (fix bugs found in use), **adaptive** 适应性 (change it for a new operating system, new law or new hardware), and **perfective** 完善性 (improve performance or add features users ask for).

■ The models

Model	Principle	Best when	Drawback
Waterfall	linear; finish each stage first	requirements are stable	poor at mid-project change
Iterative	repeated passes refine it	requirements emerge over time	harder to estimate
RAD	fast prototypes + user feedback	requirements change	needs user availability
Agile	short sprints, constant testing	change is constant	needs a committed customer

Choose by how **clear** and **stable** the requirements are at the start.

2 Practice

Now apply the methods above.

2.1 State the **purpose** of a development life cycle.

[1]

2.2 Name the **five** stages of the life cycle, **in order**.

[2]

2.3 At which stage are the **data structures and algorithms** decided? [1]

2.4 State what happens during the **maintenance** stage. [1]

3 Exam-style questions

3.1 Describe what happens in the **analysis** stage and in the **testing** stage. [4]

3.2 State **one** benefit of the **waterfall** model and **one** benefit of the **iterative** model. [2]

3.3 A company's project has requirements that are likely to **change a lot** during development. State a suitable life-cycle model and **justify** your choice. [3]

3.4 (a) State what **adaptive** maintenance is. [1]

(b) Give **one** example of a change that would need adaptive maintenance. [1]

4 Go further

You are now ready for the real exam questions on this subtopic. Open the **12.1 Program Development Life Cycle** past-paper sheet in the Library, or try these in **Practice mode**:

- 9618/22 J24 —Q1 (life-cycle stages)
- 9618/21 N24 —Q5 (development models)
- 9618/22 N25 —Q5 (the life cycle)
- 9618/23 N24 —Q5 (program maintenance)

Solutions

2.1 To **plan, manage and control** the project —to build the right product, on time, to good quality.

2.2 Analysis → Design → Coding → Testing → Maintenance.

2.3 The **design** stage.

2.4 Keeping the released program **working and useful** —fixing faults found in use, and adapting it to new needs.

3.1 Analysis —find and document **what** the program must do (the requirements). **Testing** —run the program against **test data**, compare with expected results, and fix the bugs found.

3.2 Waterfall —any one: clear/simple, well documented, easy to manage with stable requirements. Iterative —any one: problems caught earlier, copes with changing requirements, a working partial version appears sooner.

3.3 A suitable model: **agile** (or iterative / RAD), because it works in **short repeated cycles with user feedback**, so changing requirements can be handled without restarting the whole project.

3.4 (a) Changing the program so it still works in a **new environment** —a new operating system, new hardware, or a change in the law. **(b)** e.g. the OS is upgraded; a new tax rate / data-protection law; the program must run on new hardware.