

A-Level Computer Science

Name: _____ Score: _____

1. A development life cycle is used to:
 - ☐ plan, manage and control a software project
 - ☐ write the code automatically
 - ☐ replace testing
 - ☐ remove the need for design
2. A structure chart shows:
 - ☐ how a program is decomposed into modules and the parameters passed between them
 - ☐ the states a system can be in
 - ☐ the order of SQL statements
 - ☐ the layout of the database tables
3. Designing test cases from the specification only (inputs and expected outputs), ignoring the code inside, is called _____-box testing.

4. The waterfall model is best suited to projects where:
 - ☐ the requirements are stable and well understood from the start
 - ☐ requirements change constantly
 - ☐ the customer wants to see versions early and often
 - ☐ no documentation is wanted
5. A state-transition diagram shows:
 - ☐ the states a system can be in and the events that move it between them
 - ☐ the modules of a program
 - ☐ the variables and their types
 - ☐ the test data
6. For a field that accepts marks 0–100, which are the boundary test values?
 - ☐ 0, 100, and just outside (-1, 101)
 - ☐ 50 and 75
 - ☐ only "abc"
 - ☐ any random numbers

7. In the waterfall model each stage is finished before the next begins, whereas agile works in short iterative sprints with constant feedback.
- ☐ True ☐ False
8. A state-transition diagram makes missing or unhandled transitions easy to spot, because every state and the events between them are laid out.
- ☐ True ☐ False
9. Corrective maintenance fixes faults, perfective maintenance improves features, and adaptive maintenance keeps the software working in a changed environment.
- ☐ True ☐ False
10. The analysis stage is mainly about deciding:
- ☐ what the program must do (the requirements)
 - ☐ how to store the data
 - ☐ which programming language to type in
 - ☐ how to advertise the product

1. **plan, manage and control a software project**

It structures the work into stages so the team builds the right product, on time, with quality.

2. **how a program is decomposed into modules and the parameters passed between them**

A structure chart is the hierarchical breakdown into modules, with parameters down and results up.

3. **black**

Black-box tests from the spec; white-box uses the code's internal structure to cover statements and branches.

4. **the requirements are stable and well understood from the start**

Waterfall is linear with each stage finished before the next — great for stable requirements, poor at mid-project change.

5. **the states a system can be in and the events that move it between them**

States are circles; transitions are event-labelled arrows. Ideal for vending machines, locks, traffic lights.

6. **0, 100, and just outside (-1, 101)**

Boundary data sits at the edges of the valid range (and just outside) — where off-by-one errors hide.

7. **True**

Waterfall is linear (good for stable requirements); agile adapts through short sprints with continuous customer collaboration and testing.

8. **True**

Seeing every state and event reveals cases you have not handled — e.g. an unexpected second coin in a vending machine.

9. **True**

Three maintenance types: corrective (fix bugs), perfective (enhance), adaptive (new OS/hardware/rules).

10. **what the program must do (the requirements)**

Analysis gathers requirements (the "what"); design decides the "how".