12. Software Development

Starter Quiz ·

A-Level Computer Science Name: _____ Score: _____

1. A development life cycle is used to:

   ☐ replace testing

   ☐ plan, manage and control a software project

   ☐ remove the need for design

   ☐ write the code automatically

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 layout of the database tables

   ☐ the order of SQL statements

3. A program divides a total by the number of entries and crashes when the input file is empty. This is a:

   ☐ boundary error

   ☐ syntax error

   ☐ logic error

   ☐ run-time error

4. Which are purposes of a development life cycle? Select **all** that apply.

   ☐ tracking progress against milestones

   ☐ managing complexity by dividing the work into phases

   ☐ building testing into the project rather than leaving it to the end

   ☐ guaranteeing that the finished program has no errors

   Tick every correct option.

5. Breaking a problem into modules from the top down is called:

   ☐ stepwise (top-down) design

   ☐ hashing

   ☐ object code

   ☐ linear search

6. Designing test cases from the specification only (inputs and expected outputs), ignoring the code inside, is called _____-box testing.

   _____

7. The waterfall model is best suited to projects where:

   ☐ the customer wants to see versions early and often

   ☐ no documentation is wanted

   ☐ requirements change constantly

   ☐ the requirements are stable and well understood from the start

8. On a structure chart, a small arrow pointing down from the caller into a module shows a _____ passed to it.

   _____

9. A test plan lists, for each test, the input data and the expected output.

   ☐ True ☐ False

10. In the waterfall model the customer sees a working version of the program early in the project.

    ☐ True ☐ False

12. Software Development

Answers ·

A-Level Computer Science

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. **run-time error**

   Division by zero happens while the program runs and stops it. A guard such as IF Count > 0 fixes it.

4. **tracking progress against milestones; managing complexity by dividing the work into phases; building testing into the project rather than leaving it to the end**

   A life cycle plans, manages and controls. It cannot guarantee correctness; it makes finding and fixing errors part of the plan.

5. **stepwise (top-down) design**

   Top-down design produces a modular solution.

6. **black**

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

7. **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.

8. **parameter**

   Downward arrows are parameters going in; upward arrows are results returned. Together they give the module's header.

9. **True**

   Purpose, input, expected output and a column for the actual output. The strategy is the high-level approach; the plan is the detailed list.

10. **False**

    Nothing runs until coding and testing, near the end. Seeing working software early is the iterative model's and RAD's strength.