

A-Level Computer Science

Name: _____ Score: _____

1. Imperative (procedural) programming is based on:
 - ☐ a sequence of commands that change the program's state
 - ☐ stating facts and rules for an engine to infer
 - ☐ objects that combine data and methods
 - ☐ pure functions with no side effects
2. An object combines:
 - ☐ attributes (data) and methods (operations)
 - ☐ only data
 - ☐ only functions
 - ☐ two separate programs
3. What does opening a file FOR WRITE do to existing contents?
 - ☐ it overwrites them (the file is created/emptied)
 - ☐ it adds to the end
 - ☐ it leaves them and reads only
 - ☐ it encrypts them
4. An exception is:
 - ☐ an error or unexpected condition that occurs while a program runs
 - ☐ a syntax mistake caught at compile time
 - ☐ a comment in the code
 - ☐ a type of variable
5. Declarative programming means you specify:
 - ☐ what to compute, leaving the runtime to work out how
 - ☐ every step the machine must take
 - ☐ the exact registers to use
 - ☐ the order of machine instructions
6. Declarative paradigms (functional, logic, SQL) state WHAT to compute and let the runtime decide how, whereas imperative code spells out every step.
 - ☐ True ☐ False

7. Inheritance models an "is-a" relationship (a Manager is an Employee), so a subclass inherits and extends its superclass.
- ☐ True ☐ False
8. Forgetting to close a file can lose buffered writes and lock other programs out — so you should always close a file when finished.
- ☐ True ☐ False
9. Silently "swallowing" an exception (catching it but doing nothing) hides real errors and makes debugging hard — you should handle it or at least log it.
- ☐ True ☐ False
10. A constructor is a special method that:
- ☐ runs when an object is created, to set up its attributes
 - ☐ deletes an object
 - ☐ copies a file
 - ☐ runs every loop

1. a sequence of commands that change the program's state

Imperative code gives step-by-step commands (assignments, loops, calls) that change state.

2. attributes (data) and methods (operations)

An object bundles its data (attributes) with the operations (methods) that act on it.

3. it overwrites them (the file is created/emptied)

WRITE creates or overwrites the file. To keep existing contents and add more, use APPEND.

4. an error or unexpected condition that occurs while a program runs

Exceptions are run-time problems (divide by zero, file not found) that handling lets you respond to gracefully.

5. what to compute, leaving the runtime to work out how

Declarative code (functional, logic, SQL) states the goal; the runtime decides the steps.

6. True

A SQL query says which rows you want, not how to scan the tables — the opposite of step-by-step imperative code.

7. True

Inheritance is "is-a" specialisation; "has-a" relationships are modelled by composition instead.

8. True

Closing flushes buffered data to disk and releases the lock — a FINALLY block is a good place to guarantee it.

9. True

An empty handler hides the very problems you need to find; always respond or record the error.

10. runs when an object is created, to set up its attributes

The constructor initialises a new object's attributes at creation time.