

1.2 SOFTWARE & SOFTWARE DEVELOPMENT · 1.2.4 · FULL A-LEVEL

Object-oriented programming concepts

Classes, objects, **encapsulation**, **inheritance** and **polymorphism**, in OCR pseudocode. Spec 1.2.4(e).

01 The vocabulary

Class	Blueprint: attributes + methods.
Object	An instance of a class, with its own data.
Attribute	A variable storing the object's data.
Method	A subroutine defined in the class.
Constructor	<code>public procedure new(...)</code> sets attributes; create with <code>x = new Dog("Rex")</code> .

02 The three pillars

Encapsulation **Private** attributes + public get/set; class controls its own data.

Inheritance Subclass inherits superclass's attributes/methods; `super.new(...)`.

Polymorphism Same method name, different behaviour — subclass **overrides**.

03 The class pattern to memorise (OCR pseudocode)

Skeleton `class Dog → private name → public procedure new(givenName) ... endprocedure → public function getName() return name endfunction → endclass`

Validate in

set e.g. `setAge` only assigns if `newAge >= 0` — that's encapsulation earning its marks.

Inherit `class Puppy inherits Dog ... call super.new(...)` in the constructor.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Seven slips on OOP questions — one of Paper 1's most-examined topics.

01

Class = blueprint; **object** = instance. Get this distinction word-perfect.

02

Encapsulation = **private attributes** + **public get/set** — and say **why** (data can't be made invalid).

03

Class-writing questions need the **constructor** (new) and **private** on attributes.

04

Inheritance: the **subclass** inherits from the **superclass** — and can add or **override**.

05

Polymorphism = same method name, **different behaviour** in different classes (overriding).

06

Creating an object: `myDog = new Dog("Rex", 3)` — a one-line question that comes up often.

07

For "OOP vs procedural" discussions: inheritance (reuse), encapsulation (safe data), polymorphism (uniform handling), independent testing — applied to the scenario, with a justified conclusion.