

2.2 PROGRAMMING TECHNIQUES · 2.2.1(F)

OOP techniques

Classes, objects, encapsulation, inheritance & polymorphism. Spec 2.2.1(f).

01 Classes & objects

Class **Blueprint:** attributes + methods.

Object An **instance** of a class.

Instantiate `rex = new Dog("Rex")`

Constructor `new(...)` sets up a new object.

02 Encapsulation

Idea Bundle data + methods; keep data **private**.

Access Via **get/set** methods only.

Why Protects data · allows **validation** in setters.

03 Inheritance

What Child **inherits** parent's attributes & methods.

Relation "is-a" (Puppy inherits Dog).

Benefit Define once in parent → reuse, no repetition.

04 Polymorphism

What Same method call → behaves differently per class.

How Child **overrides** a parent method.

Use Loop over mixed objects, call one method.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on OOP questions.

01

Class = blueprint; **object** = instance. Dog vs rex .

02

Encapsulation = attributes **private**, reached via **get/set**.

03

Child **inherits** parent (not the reverse).

04

Polymorphism = same call, different behaviour (override).

05

Write a constructor: `new( . . . )` sets the private attributes.

06

"OOP vs procedural" → benefits of OOP **and** when procedural is simpler.