NAME DATE STARTED TARGET END DATE

GCE AS LEVEL — COMPUTER SCIENCE UNIT 1 QUESTION PACK

2500U10-1 · 2015 spec Unit 1 Topic 7 · AS unit, first sat 2017, 100 marks, 2h paper

REVISE.wales

COMPUTER SCIENCE — UNIT 1 · Programming Concepts — OOP, Parameters & Paradigms

Topic 1.4 — Programming paradigms, procedure and function parameters, object-oriented design

Procedural, object-oriented, declarative and event-driven paradigms, passing parameters by reference and by value, the relationship between objects, classes and methods, and reading UML-style class diagrams.

2015 specification · current

Estimated time for entire question pack: ~38 min

Derived from the Unit 1 pace of ~1.2 min/mark, padded for written-prose answers (25 marks over 4 questions).

*You are advised to **not** attempt to complete all of this in one sitting.*

ABOUT THIS QUESTION PACK

This is a **comprehensive topic question pack**, not a single mock paper. It contains every question from the WJEC AS Unit 1 papers (Summer 2017 – Summer 2024, COVID gap) that maps onto Topic 1.4 of the 2015 specification.

Questions are ordered by source paper date.

INSTRUCTIONS

Use black ink or black ball-point pen. Show all working. A calculator is allowed where useful.

For Examiner's use only

Q	Source	Max	Mark
1	S17 Q9	4	
2	S19 Q6	5	

Q	Source	Max	Mark
3	S22 Q11	8	
4	S24 Q10	8	
Total		**25**	

Programming Concepts — OOP, Parameters & Paradigms — what the spec asks

WJEC GCE AS Computer Science (from 2015) · Unit 1: Fundamentals of Computer Science · Topic 1.4.

Programming paradigms

- Procedural: sequence of procedures operating on shared data.
- Object-oriented (OOP): code organised into objects with state + behaviour.
- Declarative: describe what (not how); e.g. SQL, Prolog, functional.
- Event-driven: code runs in response to events (UI clicks, sensors).

OOP fundamentals

- Class: blueprint of attributes (state) and methods (behaviour).
- Object: instance of a class.
- Encapsulation: hide internal state; expose via methods (getters / setters).
- Inheritance: subclass extends superclass; polymorphism: same call, different behaviour.

Class diagrams

- Class box has three sections: name, attributes, methods.
- Visibility prefixes: + public, - private, # protected.
- Association lines show relationships; arrowheads show direction.
- Inheritance: hollow triangle from subclass to superclass.

Parameter passing

- By value: caller's variable copied; changes inside procedure don't affect caller.
- By reference: caller's variable shared; changes inside affect caller.
- Use by-reference for large structures (avoid copy cost) or to return multiple values.
- Use by-value for read-only data — safer, easier to reason about.

Variable scope

- Local: declared inside procedure / function; only visible there.
- Global: declared at program level; visible everywhere.
- Avoid globals where possible — they create hidden coupling.
- Same name can be reused in different scopes (shadowing).

Procedures vs functions

- Procedure: performs an action; may not return a value.
- Function: returns a value; ideally has no side effects.
- Both improve modularity, reuse and readability.
- Recursion: procedure / function that calls itself with smaller input.

Crib sheet »

Programming Concepts — OOP, Parameters & Paradigms in one page

Quick-reference notes — revisit before each question.

Paradigm one-liners

Procedural: step-by-step.
OOP: objects with state + behaviour.
Declarative: what, not how.
Event-driven: react to events.

Class vs object

Class: blueprint (e.g. Car).
Object: instance (e.g. `myCar = new Car("red")`).
Methods belong to class; state belongs to objects.

Pass by value vs ref

By value: copy — caller's var unchanged.
By reference: alias — caller's var changed.
Use ref for large structs or multiple returns.

Encapsulation

Hide internals; expose methods.
Reduces coupling; can change internal representation without breaking callers.

Inheritance

Subclass 'is-a' superclass.
Reuses parent attributes / methods; overrides where needed.
Polymorphism: parent reference, child behaviour.

Recursion

Function calls itself with smaller input.
Base case stops the recursion.
e.g. factorial(n) = n × factorial(n−1), base factorial(0) = 1.

Examiner only

9. Describe the object-oriented approach to programming and the relationship between an object, class and method. [4]

Examiner only

6. Parameters can be passed to procedures by reference.

(a) Explain what is meant by the term parameter and how passing parameters by reference works. [2]

(b) Describe another method for passing parameters to a procedure. [2]

(c) Give **one** disadvantage of passing parameters by reference. [1]

Examiner only

11. Describe the distinguishing features of the following different types of programming paradigms.

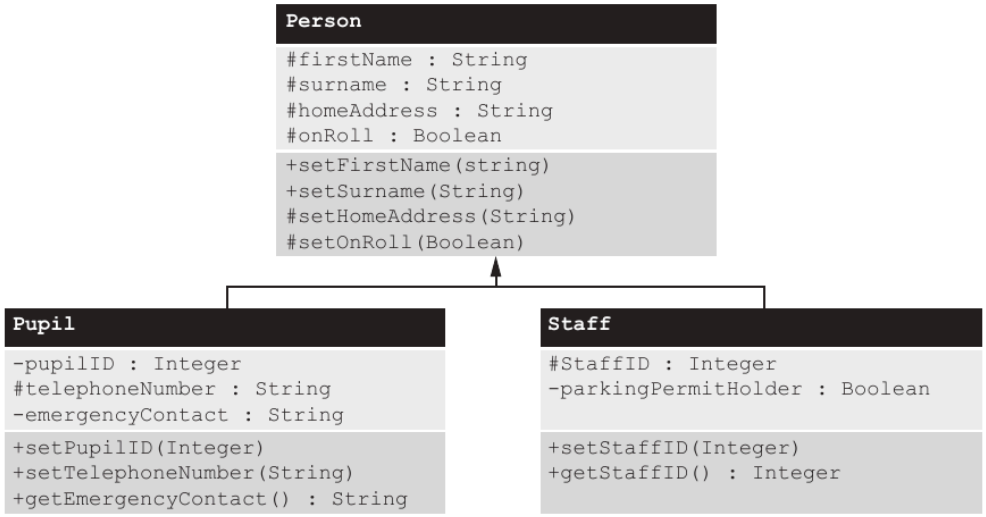
(a) Procedural. [4]

(b) Mark-up. [4]

Examiner only

10. Object-oriented programming is a paradigm that organises code into reusable, modular objects that can interact with each other to accomplish specific tasks.

Consider the following class diagram for a school:



(a) Describe the relationship between an object and a class. [2]

Examiner only

(b) Give **one** example for each of the following from the class diagram opposite.

(i) Super class. [1]

(ii) Sub class. [1]

(iii) Method that does not require a parameter. [1]

(c) Describe the difference between public, private and protected methods. [3]

END OF QUESTION PACK

4 questions · 25 marks · ~38 min
Source: WJEC AS Computer Science Unit 1 (2500U10-1), Summer 2017–2024, COVID gap
Curated for WJEC Computer Science 2015 spec AS Unit 1 — Topic 7 (1.4)