NAME DATE STARTED TARGET END DATE

GCE AS LEVEL — COMPUTER SCIENCE UNIT 1 QUESTION PACK

2500U10-1 · 2015 spec Unit 1 Topic 8 · AS unit, first sat 2017, 100 marks, 2h paper

REVISE.wales

COMPUTER SCIENCE — UNIT 1 · System Software & Development Tools

Topic 1.6 — Operating systems, utilities, translators, IDE, modes of operation and software development

Role of the operating system in managing hardware and providing a user interface, utility software, high- versus low-level languages, the compilation process (lexical, syntactic, semantic analysis), IDE features, modes of operation (batch, online, real-time), maintenance, documentation, and bespoke vs off-the-shelf software.

2015 specification · current

Estimated time for entire question pack: ~2 h 20 min

Derived from the Unit 1 pace of ~1.2 min/mark, padded for written-prose answers (93 marks over 12 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 1.6 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 Q11	8	
2	S17 Q14	12	
3	S18 Q11	11	
4	S18 Q12	8	
5	S18 Q13	11	
6	S19 Q12	6	

Q	Source	Max	Mark
7	S22 Q8	4	
8	S23 Q13	3	
9	S24 Q11	8	
10	S24 Q12	6	
11	S24 Q13	6	
12	S24 Q14	10	
Total		**93**	

System Software & Development Tools — what the spec asks

WJEC GCE AS Computer Science (from 2015) · Unit 1: Fundamentals of Computer Science · Topic 1.6.

Operating system role

- Manages hardware resources (CPU, RAM, I/O devices).
- Provides user interface (CLI, GUI, touch).
- Process / memory / file management.
- Security and access control between users and processes.

Utility software

- Antivirus, backup, defragmenter, compression tools, file managers.
- Disk cleanup, registry editors, system monitors.
- Network utilities (ping, traceroute), driver managers.
- Often bundled with OS, but distinct from application software.

High vs low level languages

- Low-level: assembly, machine code; close to hardware, fast, harder to write.
- High-level: Python, Java, C#; readable, portable, slower per operation.
- Translators bridge the gap: assembler, compiler, interpreter.
- Modern languages compile to bytecode or native machine code.

Compilation stages

- Lexical analysis: source → tokens (keywords, identifiers, literals).
- Syntactic analysis (parsing): tokens → abstract syntax tree (checks grammar).
- Semantic analysis: type checking, scope resolution.
- Code generation + optimisation → machine code or intermediate code.

IDE features

- Editor with syntax highlighting and auto-complete.
- Built-in debugger: breakpoints, step-through, variable watch.
- Compiler / interpreter integration; run output in same window.
- Project management, version control, refactoring tools.

Software lifecycle artefacts

- Bespoke: custom-built for one client; expensive but exact fit.
- Off-the-shelf: mass-market; cheap, instant, but compromise on fit.
- Maintenance: corrective (fix bugs), adaptive (new environment), perfective (improve), preventive.
- Documentation: user manuals + maintenance / technical docs for future developers.

Crib sheet »

System Software & Development Tools in one page

Quick-reference notes — revisit before each question.

OS responsibilities

Process scheduling.
Memory management (allocation, virtual memory, paging).
File system.
I/O via device drivers.
Security & user management.

Translators

Assembler: assembly → machine code.
Compiler: source → machine code (whole program, then run).
Interpreter: line-by-line execution — slower, but easier debugging.

Compilation stages

Lexical → tokens.
Syntactic → parse tree.
Semantic → type-check.
Code generation → machine code.
Optimisation throughout.

Modes of operation

Batch: process queued jobs unattended (payroll).
Online: interactive (web booking).
Real-time: hard deadlines (flight control).
Time-sharing: multi-user illusion.

Maintenance types

Corrective: fix defects.
Adaptive: new environment (new OS).
Perfective: enhance.
Preventive: pre-empt future issues.

Documentation

User docs: how to operate the system — tutorials, FAQ, manuals.
Maintenance / technical docs: for future devs — design notes, API, build instructions, test plans.

Examiner only

11. *(a)* The operating system enables the user to set up a hierarchical storage structure.

(i) Draw a clearly labelled diagram to illustrate a hierarchical structure. [1]

(ii) Give one advantage of using this structure. [1]

.....

.....

(b) Attributes can be assigned to files by the operating system. For example a read only file cannot be altered.

Name and describe three other file attributes. [6]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Examiner only

14. Describe the difference between high and low level languages.

Explain the role of the Integrated Development Environment (IDE) in developing high and low level language programs. [12]

Examiner only

END OF PAPER

Examiner only

11. Operating systems manage computer resources and provide users with a range of utility software.

(a) Explain the use of a range of utility software in computer systems. [5]

(b) Explain how an operating system manages computer resources. [6]

Examiner only

12. Compare bespoke and off-the-shelf software. [8]

Examiner only

13. An analyst has been commissioned to produce a new computer based system for weather forecasting.

Discuss the purpose of a feasibility study and describe the processes that an analyst would carry out during a feasibility study.

Describe the role of the computer in weather forecasting. [11]

Examiner only

END OF PAPER

Examiner only

(c) Give the advantages of representing numbers in integer form and in floating-point form. [4]

12. Describe the different methods of investigation used by a systems analyst. [6]

Examiner only

8. Describe a suitable mode of operation for producing payslips in a payroll application. [4]

Examiner only

13. Describe, giving an example, **one** type of maintenance performed on a computer program. [3]

Examiner only

11. Describe the contents and use made of the following documentation:

(a) User documentation. [4]

(b) Maintenance documentation. [4]

Examiner only

12. Explain the role of Integrated Development Environment (IDE) tools in developing and debugging programs. [6]

Examiner only

13. Describe the following stages in the compilation process:

(a) Lexical analysis. [3]

(b) Semantic analysis. [3]

Examiner only

14. Describe the role of the operating system in providing a user interface and managing hardware resources. [10]

Examiner only

END OF PAPER

END OF QUESTION PACK

12 questions · 93 marks · ~2 h 20 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 8 (1.6)