GCE Examiners' Report

Computer Science

A Level

Summer 2024

Introduction

Our Principal examiners' report provides valuable feedback on the recent assessment series. It has been written by our Principal Examiners and Principal Moderators after the completion of marking and moderation, and details how candidates have performed in each unit.

This report opens with a summary of candidates' performance, including the assessment objectives/skills/topics/themes being tested, and highlights the characteristics of successful performance and where performance could be improved. It then looks in detail at each unit, pinpointing aspects that proved challenging to some candidates and suggesting some reasons as to why that might be.[1]

The information found in this report provides valuable insight for practitioners to support their teaching and learning activity. We would also encourage practitioners to share this document – in its entirety or in part – with their learners to help with exam preparation, to understand how to avoid pitfalls and to add to their revision toolbox.

Further support

Document	Description	Link
Professional Learning / CPD	WJEC offers an extensive programme of online and face-to-face Professional Learning events. Access interactive feedback, review example candidate responses, gain practical ideas for the classroom and put questions to our dedicated team by registering for one of our events here.	https://www.wjec.co.uk/home/professional-learning/
Past papers	Access the bank of past papers for this qualification, including the most recent assessments. Please note that we do not make past papers available on the public website until 12 months after the examination.	Portal by WJEC or on the WJEC subject page
Grade boundary information	Grade boundaries are the minimum number of marks needed to achieve each grade. For unitised specifications grade boundaries are expressed on a Uniform Mark Scale (UMS). UMS grade boundaries remain the same every year as the range of UMS mark percentages allocated to a particular grade does not change. UMS grade boundaries are published at overall subject and unit level. For linear specifications, a single grade is awarded for the subject, rather than for each unit that contributes towards the overall grade. Grade boundaries are published on results day.	For unitised specifications click here: Results, Grade Boundaries and PRS (wjec.co.uk)

[1] ***Please note that where overall performance on a question/question part was considered good, with no particular areas to highlight, these questions have not been included in the report.***

Exam Results Analysis	WJEC provides information to examination centres via the WJEC Portal. This is restricted to centre staff only. Access is granted to centre staff by the Examinations Officer at the centre.	Portal by WJEC
Classroom Resources	Access our extensive range of FREE classroom resources, including blended learning materials, exam walk-throughs and knowledge organisers to support teaching and learning.	https://resources.wjec.co.uk/
Bank of Professional Learning materials	Access our bank of Professional Learning materials from previous events from our secure website and additional pre-recorded materials available in the public domain.	Portal by WJEC or on the WJEC subject page.
Become an examiner with WJEC.	We are always looking to recruit new examiners or moderators. These opportunities can provide you with valuable insight into the assessment process, enhance your skill set, increase your understanding of your subject and inform your teaching.	Become an Examiner | WJEC

Contents

	Page
Executive summary	5
Unit 1 – Fundamentals of Computer Science	7
Unit 2 – Practical Programming to Solve Problems	10
Unit 3 – Programming and System Development	12
Unit 4 – Computer Architecture, Data, Communication and Applications	15
Unit 5 (NEA) – Programmed Solution to a Problem	18
Supporting you – useful contacts and links	22

Executive Summary

The 2024 Computer Science GCE Assessments assessed candidates across a wide range of content and skills, including programming, system development, computer architecture, data representation, and practical problem-solving. While overall performance was reasonably good, there were notable variations between units and topic areas.

Unit 1 contributed 25% of the overall A Level and focused on fundamentals like logic, algorithms, data representation, programming concepts, and systems software. The mean score of 46.5/100 represented a decrease from 2023, suggesting candidates found this paper more challenging. Areas of relative strength included Boolean algebra and describing typical network packet contents. However, shortcomings were evident in truth tables, converting between number representations, understanding fixed vs variable length records, and describing file access methods. Candidates also struggled with advanced programming topics like object-oriented principles, documentation types, the compilation process, and the role of operating systems.

The practical programming assessment in Unit 2 saw many candidates well-prepared, especially for elements like file handling, GUI development, and authentication concepts. However, implementing thorough validation and working with provided class/form setup code proved problematic for some candidates.

Unit 3 covered programming and system development. Here performance was mixed - candidates tended to do well on topics allowing straightforward application, like simplifying Boolean expressions, writing algorithms, and truth tables. But they had shortcomings with descriptive questions requiring reasoning and examples, such as standardisation challenges, version management, sorting algorithm recursion, and natural language processing.

Outcomes for the theory-focused Unit 4 examination aligned with previous series. Practiced skills like number representation, hashing algorithms, parallelisation calculations, and assembly language saw good performance. But open-ended items demanding extended reasoning, like those covering interrupts, DBMS purposes, cloud storage trade-offs, database design criteria, and Big-O analysis proved very challenging.

The NEA project - Unit 5 also highlighted programming skills. Stronger projects exhibited comprehensive investigations, clear design documentation, rigorous testing, and insightful evaluations, while shortcomings included incomplete specifications, lack of advanced data structures, and limited self-analysis.

Some overarching areas for improvement included:

- Thoroughly describing concepts and applying them to relevant examples instead of vague statements
- Providing clear reasoning, justifications, and working instead of just stating facts
- Careful reading to fully address all aspects of questions
- More practice with open-ended, evaluative questions requiring analysis

Areas for improvement	Classroom resources	Brief description of resource
General	Portal (WJEC)	Past papers, marking schemes and exemplar materials
General – administration	WJEC website	Specification
General – administration	WJEC website	Teacher guidance
Units 1, 2, 3 & 4	Digital Resources (wjec.co.uk)	Knowledge organisers
Units 1, 2, 3, 4 & 5	Ada Computer Science	Theory website

COMPUTER SCIENCE

GCE

Summer 2024

UNIT 1 – FUNDAMENTALS OF COMPUTER SCIENCE

Overview of the Unit

Unit 1 contributes 62.5% of the AS Level qualification and 25% of the A Level. assesses a range of assessment objectives (AO) as follows:

- **AO1**
  - Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.

- **AO2**
  - Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms.

- **AO3**
  - Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.

During this series, a wide range of content was assessed, including:

- Truth tables
- Typical contents of a packet and network collisions
- Boolean algebra
- Data representation and data types
- Writing algorithms to solve non-standard problems
- Fixed and variable length records
- Sequential and indexed sequential file access
- Processes that protect the security of data
- Algorithm constructs
- Object-oriented programming
- User and maintenance documentation
- Integrated development environment tools
- Compilation process
- Operating systems.

The mean total for Unit 1 this series was approximately 46.5 out of 100 marks. This represents a decrease when compared with the mean total of 49 out of 100 in 2023, with candidates finding the paper a little more demanding overall.

Comments on individual questions/sections

Q.1 Around half of candidates were able to draw a truth table for the Boolean expression. A significant number of candidates gain zero marks for the question as they failed to list all correct inputs for P, Q and R.

Q.2 The majority of candidates were able to describe the typical contents of a TCP/IP packet. Popular answers included source address, destination address and the data itself.

Many candidates were able to explain network collisions and how they are dealt with. However, only a few were able to explain how they are detected.

Q.3 A high majority of candidates were able to simplify the Boolean expression.

Q.4 A minority of candidates were able to describe the term 'word length' in relation to a CPU.

Many candidates were able to convert the two denary numbers into binary and add them together. This was the third best answered question of the assessment.

A minority of candidate were able to demonstrate truncation to one binary place. A significant number of candidates took one binary place to mean truncating at the binary point `101`.

Q.5 A majority of candidates were able to write an algorithm that met the requirements of the builders' merchant. Candidates were generally strong at declaring variables, looping through values input, inputting the length and conversion method. They were also able to convert and output the correct length. Fewer candidates were able to carry out a suitable validation check with suitable error message and terminate the loop using a rogue value.

Q.6 A minority of candidates were able to describe the difference between fixed and variable length records. In particular, a significant number of candidates failed to give an example of data that could be contained in each.

Q.7 Only a few candidates were able to explain the difference between sequential and indexed sequential file access. This was the worst answered question of the assessment and nearly a tenth of candidates did not attempt it.

Q.8 A minority of candidates were able to describe computer-based processes that protect the security of data. Popular answers included password protection, firewalls and antivirus software.

Q.9 Many candidates were able to identify and describe the purpose of a constant. A majority were able to identify and describe the purpose of a variable and repetition. Around half of candidates were able to identify and describe the purpose of selection.

Q.10 A minority of candidates were able to describe the relationship between an object and a class.
Many candidates were able to give an example of a super class and sub class. These were the best answered questions of the assessment.

A few candidates were able to give an example of a method that does not require a parameter.
Around a quarter of candidates were able to describe the difference between public, private and protected methods.

Q.11 A small minority of candidates were able to describe the contents and use made of user documentation. Only a few were able to for the maintenance documentation, and this was the second worst answered question of the assessment.

Q.12 Around half of candidates were able to explain the role of Integrated Development Environment (IDE) tools in developing and debugging programs. Popular answers included editor, statement completion and automatic formatting.

Q.13 A minority of candidates were able to describe the lexical analysis stage in the compilation process. It is worth noting that almost a sixth of candidates did not attempt this question.

Only a few candidates describe the semantic analysis stage in the compilation process. It is worth noting that almost a fifth of candidates did not attempt this question. This is the third worst answered question of the assessment.

Q.14 A minority of candidates were able to describe the role of the operating system in providing a user interface and managing hardware resources.

These candidates gave a written response that had an adequate line of reasoning with elements of coherence, relevance, and logical structure. They showed adequate understanding of the requirements of the question and a satisfactory knowledge as specified in the indicative content; providing four to seven points on operating systems from resources and / or interface as signalled in the indicative content. Their descriptions presented a discussion with limited examples, and they generally used appropriate technical terminology referring to the indicative content.

COMPUTER SCIENCE

GCE

Summer 2024

UNIT 2 – PRACTICAL PROGRAMMING TO SOLVE PROBLEMS

Overview of the Unit

Unit 2 practical programming assesses students' ability in AO2 and AO3 topics. The majority of the unit looks at programming a solution to a scenario given in the brief.

Most of the candidates demonstrated a good understanding of the specification. Many candidates were well prepared, and many excellent answers were evident. There was evidence also that some candidates had been well prepared for the majority of the practical programming elements found in section B, however, performance when using class/form setup code was surprisingly disappointing.

It has been requested by examiners that candidates write their name and candidate number in the answer document. Possibly in the header. It would also possibly help examiners if the candidate noted which language they intend to use as they often submit all three languages and not just the one, they have attempted. Or only upload the language they have attempted. It would also be suggested to centres that they MUST follow the folder naming convention in the specification. Many thanks to the vast majority of centres who do this already.

Comments on individual questions/sections

Section A:

Q.1 **(a)** This was well answered by most candidates; however, some candidates could not identify the relationships.

(b) Many good answers were seen although some candidates did not select appropriate validation methods for different fields.

Q.2 This question was generally well answered. A majority of candidates used applied it to the scenario at times to gain the AO2 marks available.

Q.3 This question required them to apply their answer to changeover methods. Some candidates could not make a final recommendation of method.

Q.4 As is usually the case with algorithm questions, some candidates scored full marks, but a large number seemed to have no clear idea about algorithm testing.

Section B:

Q.1 This question was deemed to be slightly less accessible when set due to the increase in marks to 8. With the candidates required to fix broken code, this was done well by most candidates however not all scored full marks.

Q.2 Generally, this was not always well answered. Many candidates were unable to implement validation checks nor deal with file handling, however, more attempts were seen at this question, suggesting a slightly stronger attempt by candidates in the lower mark ranges.

Q.3 This question was generally well answered. Many candidates showed detailed annotation of their own code.

COMPUTER SCIENCE

GCE

Summer 2024

UNIT 3 – PROGRAMMING AND SYSTEM DEVELOPMENT

Overview of the Unit

Most candidates were well prepared and demonstrated an expected standard of knowledge and application required for A-Level given the candidates' distributed learning over the past two years. There was a wide range of answers with many candidates achieving higher marks in questions applying Boolean algebra and truth tables. Candidates had a strong understanding of design methodologies and changeover methods to write extensively on these points.

Comments on individual questions/sections

Q.1 Most candidates demonstrated a sound understanding of two-dimensional and three-dimensional array. Marks were lost by candidates with not being able to identify suitable examples of each. Part (b) was generally well answered with most candidates gaining full marks. Part (c) on the other hand had a range of outcomes, with many gaining the 1 mark for the assignment of the correct integer and stating the name of the array but did not correct identify the correct indices.

Q.2 This question required candidates to describe the need for standardisation of computer languages. Many candidates only identified the portability of programs and programmers, and the future maintenance of programs. Very few candidates referred to the ability to develop high-quality and secure code across different devices or support with suitable examples. In part (b) many candidates were able to describe the issue of companies having different goals and priorities, but very few were able to add any additional potential difficulties involved in agreeing and implementing standards.

Q.3 Mostly, the Boolean algebra questions were well answered, with many candidates simplifying their expressions to their simplest forms using various methods. As in the previous series, some candidates correctly applied many Boolean algebraic identities, specifically De Morgan's Law in part (a), though they arrived at incorrect answers due to misapplication at earlier stages.

Q.4 Candidates were required to write an algorithm to search an unsorted one-dimensional array of strings and replace any duplicate values with the string "X" and then output the array. Most candidates were able to achieve some marks for this question by correctly declaring variables, creating a loop and the output of an array. Some were able to correctly determine the length, and correctly check for duplication and set the value to "x". Very few correctly set the starting condition for the inner loop.

Q.5 This was a well answered question, with most candidates correctly identifying three suitable inputs to test the algorithm. Some candidates lost marks in this section for not outputting all the required variables and the expected outcome, often not including either the output for 'valid' or the correct output message. Part (b) was well answered with all most every candidate attempting the question and gaining full marks.

Q.6 Although this question was attempted by most candidates, many failed to correctly describe the criteria when evaluating computer-based solutions or to support the description with a suitable example. Generally, usability and security were the best answered, with candidates incorrectly describing performance and scalability. Candidates often confused performance with the system not crashing or lagging, rather than the speed at which it performs a given task within a suitable timeframe.

Q.7 This BNF question seemed to cause some issues. With a low facility factor. Most candidates were able to get the mark for defining the alpha and digit and "http"|"https".
Some candidates were able to define past this point using the correct BNF notations.

Q.8 The question was more inaccessible to candidates compared to other items, also with a much lower mean mark. The question asked candidates to explain programme version management. Many candidates were awarded marks stating version management can be used to track changes made to program's source code. Fewer candidates went on to fully explain the various features of version management and version management tools and their features of software such as Git and GitHub.

Q.9 The question proved demanding for many candidates as it required candidates to explain and give suitable examples of recursive and non-recursive sorting algorithms. Although many candidates were able to correctly identify Merge Sort and Quicksort as recursive and Insertion Sort and Bubble Sort as non-recursive to be awarded the two marks for naming these. There were also many candidates who also names them incorrectly getting them the wrong way round. Candidates often explained the two types of algorithms, but not in the context of a sorting algorithm, meaning fewer marks could be awarded.

Q.10 Candidates were required to demonstrate how masking could be used to determine the sign of the integer in a worked example of a given signed 8-bit integer. Many candidates attempted this question and were correctly able to demonstrate the use of AND mask with most determining the state of the sign bit as positive. Some candidates were not awarded the final mark as they had completed the task but not stated the state of the sign as positive.

Q.11 The question required candidates to evaluate an algorithm using Big O notation. The question proved to be challenging for most candidates with a low average mark. The format of this item has appeared frequently in previous papers. In part (a) some candidates who were able to identify the algorithm as linear O(n) and went on to describe and evaluate the algorithm well. Part (b) was poorly answered with very few candidates determining the correct growth rate and less attempts at this part in general. In part (c), most candidates could correctly draw both axis and the time performance.

Q.12 The question required candidates to apply their knowledge of shortest path algorithms. This was well answered with candidates providing accurate explanations of the term shortest path algorithm. However, many failed to identify that it uses a weighted network to produce costs or to provide an example such as Dijkstra's algorithm, therefore not gaining full marks for this question.

Q.13 The question has a lower mean mark than was expected. Most candidates were able to define what was meant by natural language and offered an enriched discussion on the importance of avoiding ambiguity in computer language syntax. Many candidates confused natural language syntax with natural language interfaces and therefore produced limited discussions on the interfaces themselves. high-level programming languages must be unambiguous were vague and lacked technical content. There was also limited technical content or a range of interfaces available to users.

COMPUTER SCIENCE

GCE

Summer 2024

UNIT 4 - COMPUTER ARCHITECTURE, DATA, COMMUNICATION AND APPLICATIONS

Overview of the Unit

The unit 4 examination is well established and produces consistent outcomes which result in stable grade boundaries. The examination addresses each of the three Assessment Objectives (AOs) described in the specification, as follows: -

	Description	Weighting
AO1	Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.	50/100
AO2	Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms.	38/100
AO3	Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.	12/100

Therefore, the balance between the AOs is set and adhered to, which produces a structure for a consistent assessment.

For this series the majority of candidates were well prepared and were able to demonstrate a wide knowledge of the topics in the specification, as required at A2 level, with questions requiring precise responses to programming or mathematical problems generally producing high marks.

As in previous series the questions requiring more descriptive answers often produced responses that were correct but lacked adequate reasoning or detail to gain high marks. For these types of questions, it is suggested that candidates should concentrate on the technical aspects of the topic and include examples to illustrate and help structure their responses.

Comments on individual questions/sections

Example AO1 Questions

Q.1 CPU components. Mean mark of 3.6 out of 6 with a facility factor of 57.0.

This question was intended to provide an accessible start to the paper and most candidates identified three components, as required. Some candidates lost marks by missing the required descriptions and / or by listing other components, such as RAM or main memory.

Q.5 **(c)** Interrupts. Mean mark of 0.7 out of 6 with a facility factor of 11.3.

Candidates found this to be a challenging topic, with many not referring to the FDE cycle, as required, and very few mentions of an Interrupt Service Routine in the responses seen.

Q.9 **(b)** Purpose of a DBMS. Mean mark 1.0 out of 6 with a facility factor of 16.7.

A topic which proved to be more challenging than expected. Many candidates described an automated management process and were able to identify functions such as backups and access levels, but very few mentioned a database administrator or their use of the software.

Q.12 Cloud storage. Mean mark of 3.2 out of 9 with a facility factor of 24.7.

An open topic within the context of storage, backup and online retrieval of files. The question was band marked with a range of points identified in the indicative content of the mark scheme. Most candidates were able to describe some benefits and drawbacks associated with the use of the 'the cloud', but few covered the advantages provided in terms of file synchronisation and support of collaborative working, or possible complications arising from legal and regulatory requirements.

Example AO2 Questions

Q.2 Hashing algorithm. Mean mark of 3.4 out of 6 with a facility factor of 57.0.

Most candidates were able to complete the required MOD calculations accurately and identify reasons why the algorithm is unsuitable. Several candidates then went on to provide a very detailed account of alternative methods for dealing with data collisions, which, although correct, exceeded the expectation, as indicated by the marks allocated to this part of the question.

Q.4 Parallel processing. Mean mark of 2.2 out of 6 with a facility factor of 46.0.

The majority of candidates were able to complete the required calculations correctly and several produced explanations based on the number of processors, but without stating that the calculations indicate that the level of parallelisation achieved in the software is the most significant factor.

Q.8 Number representation. Mean mark of 6.0 out of 10 with a facility factor of 60.0

An accessible and well-practised topic. Most candidates were able to convert between hex and binary numbers and produce a correct representation of a negative binary number in twos complement. The parts of the question which involved 16-bit floating point numbers were also well done, with many examples of full marks seen for these items.

Q.10 Database design. Mean mark of 4.7 out of 8 with a facility factor of 58.0.

Most candidates were familiar with the design work required and were able to interpret the given scenario to produce reasonable entity relationships. Data tables also tended to be well done, although the allocation of foreign keys caused some difficulties, particularly where ERDs had not been well developed.

AO3 Questions

Q.7 Assembly language. Mean mark of 4.2 out of 6 with a facility factor of 69.8.

This is a well taught and practised topic, and most candidates were able to interpret the given scenario and successfully translate the logic of the given algorithm into assembly code, using the instruction set provided.

Q11 (b) SQL commands. Mean mark of 3.7 out of 6 with a facility factor of 61.0.

This is also a well taught and practised topic, and most candidates were able to produce the required commands correctly, although, as in previous series some candidates failed to distinguish between text and numeric fields (‘…’) as required.

COMPUTER SCIENCE

GCE

Summer 2024

UNIT 5 (NEA) – PROGRAMMED SOLUTION TO A PROBLEM

Overview of the Unit

Many projects of a good standard were submitted for moderation this summer. Moderators saw some work of an excellent standard. Many centres had assessed the work accurately and had clearly explained their assessment decisions which aided the moderation process.

Tasks
Comments on tasks/questions relating to candidate performance/meeting assessment criteria

Discussion:
It is important that centres recognise the importance of the discussion section. This section provides opportunities for the candidates to present their problem situations to their teacher, peers and/or other competent third parties. Candidates should receive detailed informed feedback regarding the scope of their chosen problem and should reflect, in depth, on the discussions and feedback to allow them to firm up their ideas and ensure that unsuitable topics are revised or discarded.

This is a good opportunity for teachers to steer candidates away from unsuitable ideas that will lack the scope required to produce work to a standard and level appropriate for this qualification. Some candidates are simply describing their feedback rather than using it to reflect and refine their ideas. If needed, candidates have the option to reconsider their problem or potentially identify a different, more suitable problem situation.

Investigation:
When candidates selected appropriate real-life problems, they were presented with a chance to conduct an investigation into the existing system. Candidates were expected to identify the data collected, processed, and produced by the current system. Where this did not take place, the investigation tended to be more narrative and therefore the ability to identify suitable objectives could not take place. For students to access the higher mark bands the investigation of the existing system should be thorough.

Where candidates are not able to identify a real-life problem or have a new idea that does not have any current end users, they should carry out extended research into similar commercial systems, identifying common characteristics and should base the conclusions of their investigations on the information that they have been able to gather.

It is important that candidates produce a comprehensive working specification and that measurable objectives are set that will inform the design, prototyping, software development and testing processes. The objectives should be determined by each section of their investigation, often students will propose ideas in their research but not follow this through in the objectives.

Design:

The design stage is an opportunity to effectively plan and develop the technical elements of the functional solution before its implementation. This should be forward-thinking and include a range of elements, including:

- designs of screen layouts
- consideration of all forms of inputs and outputs required
- evidence of all data structures with methods of access
- consideration of data validation
- design of programming routines for the functional solution using a recognised convention such as pseudocode or flowcharts etc.

Candidates should be encouraged to relate all elements of their design to their project. In some centres candidates are describing abstract concepts such as validation checks and sorting constructs. These only have merit when applied to the current project and can demonstrate some value to its implementation.

Some centres are still using screen shots of already developed code as a way of considering the processing stages. This section should be completed before the prototyping and software development stage and not retrospective.

Prototype:

The prototype work is intended to allow self-reflection on the chosen method of solution and the design work. Candidates should clearly identify the sections they have chosen to include explaining their justification for these.

Candidates can demonstrate their work through a series of screen shots of the project running and demonstrating the sections identified by the candidate. Many centres have also used video evidence for this section which is to be commended as it allows candidates to thoroughly demonstrate the workings of their solutions.

This element of the prototype is intended to allow self-reflection on the chosen method of solution and the design work. Strengths and weaknesses of the current solution should be identified with some proposals for improvement, it is not appropriate to include feedback from third parties in this section of the work.

Post prototype Refinement of Design:

This part of the project is to allow candidates to alter their design after third party feedback. Candidates should evaluate their feedback and give justifications for the acceptance or rejection of the proposals.

It is important to note this should be refinements to the original designs, some centres are showing the before and after of the designs which is giving some excellent evidence for the section. There are still some centres showing the finished programmed changes to the design, this will be assessed in Software Development and hence is not appropriate for this section.

Software Development:

To access the higher mark bands all programs should fully exploit the language, using advanced programming features and well-structured data normalised to 3NF, this was not always evident.

There should be examples of effective searching and sorting at the programming end and output content as requested by the user. The system should aim to cover all the objectives identified and fully exploit the programming facilities of the language.

Many solutions had limited validation methods but did cover basic validations for some key components. Often the program contained some self-documenting identifiers, however this was not there for all variables and structures. Many centres included some detailed annotation that helped identify where the candidates had included their complexity.

Many centres provided solutions that could not be run due to additional packages/libraries requirements, in this instance these should be provided or information on these additions provided in a 'README' file and the cover sheet. It may be of benefit to provide additional video evidence of the system being used to support this section. Where centres did this, the moderation process was made easier.

Testing:

Where centres had included a testing table clearly outlining the types of testing taking place, typical, extreme and invalid, candidates produced some excellent testing evidence. Many centres have moved to using video evidence for this, linking the videos with the individual tests. Where candidates have evidence of several tests in a single video, a time reference would be useful allowing the moderator to go directly to this.

It is important that the testing work should focus on the functionality of the solution in terms of:

- Input facilities including measures to ensure reasonable data entry
- Processing facilities to ensure correct and accurate output
- Appropriate output including screen and paper-based outputs

The testing should cover each objective with data designed to measure the outcomes of the system against the desired outcome. The quality of the commentaries accompanying the testing evidence has a major role in identifying the marks to be awarded for this section of the work. Candidates should give specific suggestions to refine the system with any tests that do not successfully pass testing.

Evaluation:

It is pleasing to see many centres correctly assessing candidates' evaluations. Higher level candidates are giving detailed analysis of their performance in each section of the project and thoughtful observations about any changes to their future practices.

For candidates to access the higher mark bands they need to cover each section of the evaluation. Some candidates can find this difficult if their initial investigation is not detailed enough as these sections link together.

A few candidates were using this section to demonstrate their solution met all the original objectives, while this is useful it is not evaluative and could be included in testing. The expectation is the candidates should be self-reflective in this section and fully analyse their own performance.

Task marking
Comments on approaches to internal marking

Administration:
It would aid the moderation process if centres would ensure that candidates' work and documentation are saved with filenames that clearly identify the centre number, candidate number and candidate name. As detailed in the specification for this qualification, "For example Diane Smith (centre number 68999, candidate number 12345) would store her work in a folder named 68999_12345_SM_D. In addition, candidates should ensure that they have linked their work to the GCE Computer Science Unit 5 Task sheet (U5e). The centre should make note on the Centre Mark Sheet the nature of any assistance given and the extent to which the solution actually works as stated in the report there should also be specific reference to the assessment objectives in the comments written on the work and coversheets.

Supporting you

Useful contacts and links

Our friendly subject team is on hand to support you between 8.30am and 5.00pm, Monday to Friday.

Tel: 02920 265 401

Email: CS@wjec.co.uk

Qualification webpage: AS/A Level Computer Science (wjec.co.uk)

See other useful contacts here: Useful Contacts | WJEC

CPD Training / Professional Learning

Access our popular, free online CPD/PL courses to receive exam feedback and put questions to our subject team, and attend one of our face-to-face events, focused on enhancing teaching and learning, providing practical classroom ideas and developing understanding of marking and assessment.

Please find details for all our courses here: https://www.wjec.co.uk/home/professional-learning/

WJEC Qualifications

As Wales' largest awarding body, WJEC supports its education community by providing trusted bilingual qualifications, specialist support, and reliable assessment to schools and colleges across the country. This allows our learners to reach their full potential.

With more than 70 years' experience, we are also amongst the leading providers in both England and Northern Ireland.

WJEC
245 Western Avenue
Cardiff CF5 2YX
Tel No 029 2026 5000
Fax 029 2057 5994
E-mail: exams@wjec.co.uk
website: www.wjec.co.uk