GCSE EXAMINERS' REPORTS

GCSE (NEW)
COMPUTER SCIENCE

SUMMER 2019

Grade boundary information for this subject is available on the WJEC public website at: https://www.wjecservices.co.uk/MarkToUMS/default.aspx?l=en

Online Results Analysis

WJEC provides information to examination centres via the WJEC secure website. This is restricted to centre staff only. Access is granted to centre staff by the Examinations Officer at the centre.

Annual Statistical Report

The annual Statistical Report (issued in the second half of the Autumn Term) gives overall outcomes of all examinations administered by WJEC.

COMPUTER SCIENCE

GCSE (NEW)

Summer 2019

Unit 1 – Understanding Computer Science

General Comments

This is the first year of awarding this new reformed specification and as such, direct comparisons with previous series for this unit are not possible.

Candidates found the new Unit 1 demanding and a mean mark below 40 confirms this. Relatively few candidates achieved a mark over 75, although a few candidates scored over 90.

At a question level, it was pleasing to note that candidates performed well in some questions, but overall, nine of the twelve questions had a facility factor below 50.

Comments on individual questions/sections

Q.1 Around half of candidates were able to complete the table, indicating the lowest cost routes from node D.

Q.2 Only a minority of candidates were able to compare the CPU performance of the two computers in terms of cache size, clock speed and number of cores. The main issue observed with this question was that candidates demonstrated a lack of understanding and gave only a superficial comparison, such as one being larger / faster / more than the other. For example, a typical answer seen was that CPU 1 has a greater clock speed. Candidates failed to complete the comparison by stating that this would result in CPU 1 being able to run the fetch-decode-execute cycle faster that CPU 2 and therefore process more instructions.

A similar issue was also observed with Q2b where candidates demonstrated a lack of understanding and gave only a superficial comparison.

Q.3 This was the best answered question of the paper. A majority of candidates were able to state the logical operator that was used to produce the output in the given truth tables. Fewer candidates were able to show the Boolean expression that represents the function described by each truth table.

Q.4 Only around a third of the marks were achieved on average by candidates for the question on network protocols and network layers. Candidates found all parts of this question demanding.

Q.5 A minority of candidates were able to identify and describe three facilities provided by the IDE.

Q.6 Many candidates had difficulties with the Boolean algebra question and lacked confidence in their use Boolean identities and rules.

Q.7 Candidates performed well in the mathematical questions and were particularly confident in converting between different number counting systems.

Q.8 Around a third of marks were awarded on average for this question. For Q8bi, some candidates did not use any units in their answer.

Q.9 Most candidates were able to explain the purpose of an acceptable network use policy and give one or two examples of typical content but struggled to give more examples than this.

Q.10 This was the worst answered question of the paper with only around a quarter of the marks on average awarded. Around half of candidates were able to describe the different forms of cyberattack, but very few were able to describe the three methods of identifying vulnerabilities. In particular, candidates had difficulties differentiating between ethical hacking and its subset penetration testing.

Q.11 Remarkably, nearly a quarter of candidates did not attempt the question describing the functionality of three utility software tools. Where description were given, they often related to other features provided as standard by many operating systems, such as a text editor, which is not a utility software.

Q.12 In general, candidates showed an adequate understanding of the requirements of the question and a satisfactory knowledge of the indicative content. They addressed the question, discussing user interfaces and used appropriate technical terminology referring to the indicative content.

Summary of key points

Candidates had a reasonably good understanding of lowest costs routes, logical operators and mathematical content. They performed best in these questions.

Candidates found the questions on methods of identifying vulnerabilities and utility software difficult, and this was reflected in the low facility factor for these questions. With a quarter of candidates not attempting the utility software question, this would also suggest that centres need to spend more time teaching the topic.

COMPUTER SCIENCE

GCSE (NEW)

Summer 2019

Unit 2 - Computational Thinking and Programming

General Comments

Most of the candidates demonstrated a good understanding of the specification, however, question attempted percentages were not available this series. Many candidates were well prepared, and many excellent answers were evident. There was evidence also that most candidates had been well prepared for some of the practical programming elements.

Comments on individual questions/sections

HTML (markup language)

This was answered well by most candidates. However, a significant number of candidates found the link (using <a href=. . .) difficult. Candidates also found the <li> tag difficult. This was the first time this was assessed and contributed to the slightly increased difficulty of the paper this year. Some candidates also forgot to close the HTML tags resulting in incorrect formatting. Candidates should not use generated code or CSS as the specification states the HTML tags that should be used.

Assembly language

This was a new addition to the specification and was deemed to be a difficult question.

Algorithms

Some good answers were seen to this question, however, only a few candidates achieved full marks. Within the question marks were awarded for sections – such as inputting data, loops, conditions (if statements) and output. Of these sections most candidates had input and some outputs. Some candidates had a correct if statement. Very few candidates had a working loop. Dry running an algorithm and fixing an algorithm were also assessed for the first time.

Java programming within the Greenfoot environment

A very small number of candidates did not save the file as the required name. Populating the world proved problematic for a small number of candidates, most candidates could get an object to move around the screen. A significant number could get an object to respond to key input.

Summary of key points

The specification for this new strengthened Computer Science GCSE states in inclusive lists the HTML tags, Assembly Mnemonics, Java programming scope (within the Greenfoot environment) and a possible structure for algorithm syntax that the WJEC will always use. Whilst candidates are not limited to these (i.e. they do not have to use the algorithm syntax in the specification, nor only learn to use Java to the extents required in the specification) the WJEC will always stick to those stated in the specification.

COMPUTER SCIENCE

GCSE (NEW)

Summer 2019

Unit 3 – Software Development

General Comments

Examples of good work were seen at moderation this summer. This new specification requires candidates to undertake the one scenario that is made available to candidates. It should be noted that it is essential that candidates undertake the correct scenario for the current series.

Comments on individual questions/sections

Requirements of the scenarios

The scenario has a bullet pointed lists of requirements.
To access full marks for the implementation of the solution to the given problem, all bullet points should be covered. However, many candidates were not able to produce a solution that covered all bullet points of the scenario.
The scenario was based on the calculation of reading ages for a series of texts.
The Automated Readability Index (ARI) is a method of calculating the reading age a child would need to have to understand a piece of writing. The method uses the following calculation:

Reading age of text = $4.71\ x\ \left(\frac{Characters}{Words}\right) +\ 0.5\ x\ \left(\frac{Words}{Sentences}\right) -\ 21.43$

Reading ages are always rounded up, if the calculation came out at 10.4 the reading age would be given as 11 years old.

For the 'ARI' scenario candidates were asked to:

- Enter and store the details of the text such as title, age of reader, keywords
- Enter the content of the text
- Calculate the number of characters, words and sentences in the text
- Use the ARI to calculate the reading age of the text
- Store the calculated reading age of the text with the text details and text
- Output the intended reading age and the actual reading age of the text
- Allow other teachers to search for the text using keywords and reading age.

Most candidates could create an interface that allowed users to enter the card details. The majority of candidates allowed users to enter the customers' personal details.
Many candidates were able to produce code that could carry out the required calculations. Candidates also created validation routines. However, these routines often caused issues when running the code as the interfaces did not provide clear enough instructions for the user.

Requirements for the Report:
The specification states that the candidates should produce a report that:

- analyses the given information
- includes a design of a solution to the given problem
- programming of a solution to the given problem
- testing and refinement of the application, noting the refinements in the refinement log
- gives an evaluation of the application.

Refinement log
Candidates are required to complete a log of their activities during the twenty hour controlled assignment. Almost all candidates presented a completed log. However, a significant number of candidates submitted logs that included many copied and pasted entries. Where candidates had made effective use of their logs; entries included discussion of problems encountered and solutions to these problems. Many candidates were able to identify action points for following sessions that would enable them to make more effective use of their time.

Scope of the problem
A minority of candidates presented effective analyses of the given scenario while many either restated the problem or copied and pasted the contents of the brief. Most candidates were able to outline the objectives for their solution to the given problem.

Design of Solution
In a significant minority of instances, candidates were neither able to justify their choice of programming language using appropriate technical terminology nor relate the features and facilities of the language to their proposed method of solution.
Many candidates were able to describe some of the process stages required for their solutions in pseudo code and/or flowcharts. However, fewer candidates covered all processing stages for their proposed solution. In several cases it was not clear that this work had been completed before implementation. Retrospective designs will not be given any credit at moderation.

Effectiveness of solution
Most candidates had produced a solution that allowed the user to enter the details and contents of a text to be analysed through calculations. A large percentage of candidates developed solutions using the Python programming language, while others produced solutions using a visual programming language. The quality of the interfaces produced varied considerably.
Most solutions were modular and included the required authentication routines.

Technical quality
Many candidates produced code that was self-documenting and there were instances of the code being well structured. Where candidates had a good understanding of the language they were using, there was evidence of the use of consistent style throughout including indentation and use of white space.
In general candidates produced code that used meaningful identifiers and appropriate constants and had coded some validation routines. Many candidates included some annotation of their code with more able candidates included annotation that demonstrated their understanding of the problem and solution.

Test strategy
Most candidates were able to describe some type of testing strategy and some evaluation criteria. In future candidates may benefit from considering their objectives when describing their testing strategies and evaluation criteria, ensuring that they plan to test and evaluate against each objective.

Testing
Most candidates were able to design tests that would demonstrate the functioning of parts of their solution. However, some test plans would have benefited from focusing on the logic of the solution rather than repeatedly testing the less complex parts of the system.
In future candidates would benefit from using their objectives and success criteria as a framework for their test plans and ensure that these are met by their solution.
As solutions should be able to carry out calculations of the ARI, candidates should ensure that the data entered produces the correct result and the output is correctly formatted. The scenarios provided many opportunities for candidates to test the logic of their solutions using text-based data to produce a mathematical result to their calculations.

Summary of key points

Further development
The specification calls for candidates to:

- considers the outcomes of the testing process in terms of how well the application meets the objectives set at the beginning of the project
- describes the good features of the application and identifies areas for further development
- provides detailed suggestions for specific extensions to the application.

Many of the discussions produced by the candidates were brief and tended to be narrative rather than reflective and evaluative in nature.
Few candidates offered valid and detailed suggestions for future improvements. However, a minority were able to discuss their solutions in the light of their structure and suggest viable improvements that could be created using their chosen language.

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