GCSE EXAMINERS' REPORTS

COMPUTER SCIENCE

SUMMER 2016
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.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 1</td>
<td>1</td>
</tr>
<tr>
<td>Unit 2</td>
<td>4</td>
</tr>
<tr>
<td>Unit 3</td>
<td>5</td>
</tr>
</tbody>
</table>
COMPUTER SCIENCE
General Certificate of Secondary Education
Summer 2016
UNIT 1

General Comments

This report is produced to support teachers by indicating where responses were generally good or poor and to indicate why candidates were not gaining marks on specific questions. The report should be used in conjunction with the question paper, mark scheme and item level data.

The wording of questions is designed to suggest the level of detail required, by the use of words or phrases like name, describe, describe in detail, explain or discuss, but it is very important that candidates also pay attention to the maximum number of marks available in each part question, as stated in the rubric for each paper.

The standard of language used was generally very good. Candidates should be reminded that legible handwriting is important throughout the whole paper. Examiners are not prepared to guess the word: if it is not clear, then no mark can be awarded.

Candidates should be reminded that they need to read the instructions on the front of the answer booklet regarding the necessity to use black ink or blank ball point pens. Unsurprisingly, black pens scan more clearly than other colours.

Centres need to be reminded to use the additional blank pages in the answer booklet before completing continuation booklets.

Centres and candidates are reminded of the value of giving examples. A good example, even if not specifically requested, can often help to secure a mark when a description is lacking in clarity or detail.

Candidates could answer questions on most of the specification and seemed better prepared for this examination compared with previous examination series. For example, it was pleasing to see that candidates were able to distinguish between functions of an operating system and the features of a graphical user interface. Also many showed good technical understanding of the different types of programming errors and were able to convert to binary and hexadecimal.

The language used by most candidates was technical and precise with many candidates correctly and confidently using computer science terms.

Many candidates were obviously well prepared for this exam and excellent answers were evident with numerous candidates giving extended answers where appropriate. However, some candidates were not prepared to answer questions on all topics.
1. Generally well answered.

2. (a) Most candidates were able to name the component even if they were unable to describe the function.
   (b) Many candidates were able to give an example of a port but fewer were able to describe the purpose of the port.

3. (a) Many good answers were seen with candidates clearly giving advantages of using cloud storage to store files.
   (b) Generally well answered.

4. (a) Generally well answered.
   (b) Most candidates gained some marks on this question but few went on to score full marks. Many answers had the idea of the switch inspecting the address on the packet but few went on to describe comparing to a stored list of addresses idea.

5. (a)&(b) Most candidates scored very well on this question which required candidates to identify and describe different error types.

6. (a) This question was not well answered. Where candidates gained marks it was for the saving time to write code idea and modules in a library should have been thoroughly tested.
   (b) Some knew this while most did not.

7. (a)-(d) This question was generally well answered with most candidates gaining some marks and many going on to score high marks.

8. (a)&(b) Very well answered by majority of candidates but candidates are reminded of the importance of showing their workings. If an answer is incorrect but has the correct method then marks can be awarded.

9. Most candidates understood the concept of a truth table and scored well. However candidates should be reminded that to use the notation of 0 or 1 given in the truth table.

10. Some good answers were seen and many candidates gave four features and four suitable examples. Some candidates simply stated WIMP and others neglected to give examples and failed to gain high marks.

11. It was pleasing to see most candidates gaining marks on this question. Candidates should be reminded to only write down the actual output produced and nothing else. If it is unclear that the candidate clearly knows what is output and what is code then marks will not be awarded.

12. (a) Very good answers were evident. However the only acceptable answer is ‘real’ and candidates should be made aware of this and not use data types from a language that they might be familiar with such as ‘float’, ‘single’ etc.
   (b) Very few candidates were aware of the term local static variable.
13. Most candidates who attempted the question were able to recognise who would use a CLI but fewer were able to give three reasons why it would be suitable.

14. (a)-(c) Most candidates understood that files would be faster to upload or download. Some candidates just stated ‘saving space’ when some idea of disc storage space was required.

15. Most candidates were able to identify username and strong passwords with fewer going on to discuss permissions or levels of access. Encryption was briefly described by most but few went on to explain the process involves an algorithm and an encryption key. Backup and recovery procedures were less well answered. Candidates should be reminded of the need for clear legible hand writing in all questions but particularly this question where and extended piece of writing is required.
COMPUTER SCIENCE

General Certificate of Secondary Education

Summer 2016

UNIT 2

General Comments

As with the previous series, I am delighted to report that there were very few procedural or logistical problems. Centres generally prepared the electronic media for delivery to examiners well. However, a small number did not use the correct candidate naming convention. Centres are reminded that they should not use generic folder names such as “candidate 1” and should test that the examination accounts can properly run Greenfoot/Java before the examination. The vast majority of centres had no issues with the on-screen nature of this examination and any feedback would be welcome.

Most candidates attempted all three components of the examination. While good answers were seen for the HTML and Greenfoot components, the responses seen for the algorithm were disappointing. A small number of candidates did not attempt the algorithm question. However, those candidates who gained full marks in the question produced very elegant solutions.

Comments on Individual Questions:

1. This was answered reasonably well by most candidates. However, a significant number of candidates found the link (using \(<a href=\ldots\) difficult. 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 tags that should be used.

2. Some disappointing answers were seen to this question. Only a few candidates had a fully working algorithm. Within the question, marks were awarded for sections – such as inputting data, loops, conditions (if statements) and output. Of these sections most candidates included input and some outputs. Some candidates included a correct if statement. Very few candidates included the concept of a loop.

3. Responses here were highly variable. Sadly 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 but most candidates could get an object to move around the screen. A significant number could get the other object to respond to key input. The method for implementing movement on key press varied by centre and all working solutions were awarded credit. Some candidates could implement collision detection and trigger a sound. Very few candidates could implement the counter. However, encouragingly, some candidates fully implemented the entire scenario and compiled it to a single (executable) jar file, showing ability above and beyond what was required for full marks. The code written was generally of a very good standard demonstrating a good understanding of objects, methods and properties within Java. WJEC emphasizes that centres should ensure that the machines on which this online examination is to be taken are tested to ensure their functioning status before the examination begins. We would also like to stress that candidates should sit the exam using the version of Java & Greenfoot that they have been taught in. A very small number of centres reported that they had used a different version of Greenfoot in the exam as was used during teaching. This is a centre based issue and should be prevented wherever possible.
General Comments

Examples of good work were seen at moderation this summer. As usual, two scenarios had been made available to the candidates. The scenarios were of similar demand and each allowed candidates to gain access to the full range of marks. Candidates’ outcomes were not affected by their choice of scenario. In most cases centres made use of one or the other scenario although there were a few centres that submitted work from both scenarios.

Requirements of the scenarios.

Each 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 for their chosen scenario.

For the ‘Betty’s Best Ever Bakery’ scenario candidates were asked to:

- Input a customer’s order which may include ingredients for more than one recipe
- Calculate and display the total quantity for each ingredient
- Calculate the fewest required number of bags and boxes of each basic ingredient
- Produce and display a list of all bags and boxes of each basic ingredient to be sent to a customer

Most candidates could create an interface that allowed users to enter their requirements. The majority of candidates allowed users to enter the quantity of both types of cakes required.

Many candidates were able to list the quantity of each ingredient required and some were able to calculate a number of bags and boxes of each basic ingredient. However, very few were able to fulfil the third bullet point by listing the fewest number of bags and boxes. Many listed, for example, two small bags rather than one medium sized bag.

For the ‘The School Council’ scenario candidates were asked to:

- Input the names of the candidates standing for the School Council election
- Input the preference votes from the ballot papers
- Display all the candidates and their total number of points
- In the event of a tie, use the preference votes to identify the winner
- Display the outcome of the election

Most candidates were able to provide an interface that allowed the candidates’ names to be entered. Many candidates provided facilities that allowed at least one round of voting to take place.

The majority of candidates were able to provide some calculated outcomes from the voting and some were able to provide a list of the candidates in rank order.
Very few candidates were able to deal with the fourth bullet point to identify the winner in event of a tie. Many candidates did not attempt to cope with this requirement.

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

- Describes the design, implementation and testing of their solution e.g.
- Sequencing
- choice (if - then - else)
- iteration (loops)
- language constructs that support abstraction (typically a ‘procedure’ or ‘function’ with parameters)
- some form of interaction with the program’s environment (input/output or event-based)
- Finds and correct errors in their code
- Reflects thoughtfully on their program, including assessing its correctness and fitness for purpose; understanding its efficiency; and describing the system to others.

Design of Solution
Most candidates were able to carry out the required analysis and many were able to outline the objectives for their solution to the given problem.

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.

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.

Program Documentation
Many candidates were able to demonstrate at least a reasonable understanding of the tools and techniques they had used through annotation. However, in some instances candidates struggled to use the correct technical terminology to describe the constructs of their programs.

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.

Where solutions carry out calculations candidates should ensure that the data entered produces the correct result and the output is correctly formatted. Both scenarios provided many opportunities for candidates to test the logic of their solutions using mathematical data.
Evaluation
Many candidates did not appear to have realised the importance or the demands of this section of work. It should be noted that 12 of the 50 marks are awarded for the review section of the work.

This being almost a quarter of the total marks an extended piece of writing was required to cover the requirements of the specification.

The specification calls for an evaluation that allows the candidate to: Reflect thoughtfully on their program, including assessing its correctness and fitness for purpose; understanding its efficiency; and describing the system to others.

Many of the evaluations produced by the candidates were brief and tended to be narrative rather than reflective and evaluative in nature.

In future candidates would benefit from assessing their work against their objectives and success criteria to ensure that their discussion is correctly structured.

Few candidates offered valid and detailed suggestions for future improvements. However, a minority were able to discuss their solutions in light of their structure and suggest viable improvements that could be created using their chosen language.