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 – Understanding Computer Science</td>
<td>1</td>
</tr>
<tr>
<td>Unit 2 – Solving Problems Using Computers</td>
<td>3</td>
</tr>
<tr>
<td>Unit 3 – Developing Computing Solutions</td>
<td>5</td>
</tr>
</tbody>
</table>
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.

Most candidates demonstrated that they understood and could answer questions on most of the specification. Many candidates were obviously well prepared for this exam and excellent answers were evident with numerous candidates giving extended answers where appropriate. However it was clear that some candidates were not prepared to answer all questions.

1. Most candidates were able to identify whether memory was volatile or non-volatile although many candidates were unable to give examples of typical data that might be stored.
2. This question was not well answered as many candidates named proprietary software rather than answering the question with the generic name.
3. Most candidates were able to identify a suitable device but fewer were able to justify their choice.
4. Generally well answered as candidates were able to label components of star and bus networks. Fewer were able to give three advantages of a star topology compared with a bus topology.
5. This question was not well answered. Many candidates were not able to apply the appropriate principle of the Data Protection Act for the given scenarios. A large majority of candidates failed to give suitable reasons for insisting that employees agree to follow the company’s code of conduct and many candidates incorrectly thought that this would exonerate an organisation’s legal responsibility.
6. Generally this question was not well answered. Many candidates were unable to describe the defragmentation process and functions carried out by system restore. Most candidates were however able to state typical functions carried out by virus checker software.
7. Well answered by some candidates however it was disappointing that far too many candidates could not describe the role of a Domain Name System server and therefore could not explain how DNS servers are used to answer a request made by a web browser to display a web page.

8. Generally well answered however a large minority of candidates either did not attempt the question or could not describe the difference between the way in which a compiler and an interpreter operate.

9. Generally well answered as candidates were able to convert between denary and binary. A large majority of candidates answered 1 rather than 0 as the minimum denary number that can be represented using 8 bits.

10. 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. Fewer candidates were able correctly identify the three outputs from the algorithm.

11. Most candidates were able to convert from binary to hexadecimal. Fewer candidates were able to give two reasons for the use of hexadecimal. Many candidates did not state that a MAC address is unique.

12. Many good answers were seen however far too many candidates were only able to identify the physical resources that the operating system manage and were not able to describe its role.

13. Many candidates failed to achieve marks for this question. Candidates should be reminded to only write down the actual output produced and nothing else.

14. Most candidates were able to identify an integer as a suitable data type for a Count variable and a real for a Mean variable.

15. Some very good responses were evident as candidates were able to give three types of errors and use their theoretical and practical knowledge to identify how the errors are detected and strategies that would reduce the chance of the errors. Many candidates however were only able to identify three errors.
COMPUTER SCIENCE
GCSE
Summer 2017
UNIT 2 SOLVING PROBLEMS USING COMPUTERS

General Comments

As with the previous series, I am delighted to report that there were very few procedural or logistical problems. There was a new method for submitting candidate scripts to the WJEC (using BTL secure access) and this largely worked very well. 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 via the subject officer would be welcome.

There was no data available this series on question attempted/not attempted. While good answers were seen for the HTML and Greenfoot components, the responses seen for the algorithm were disappointing. Those candidates who gained full marks in the question produced very elegant solutions.

Unit 2: 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. Candidates also found the \(<img>\) 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 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 had input and some outputs. Some candidates had a correct if statement. Very few candidates had the concept of a loop.

3. Responses here were highly variable. 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 the spaceship 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 which is seen as an A/A* differentiator. 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. The 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 the WJEC specifies. 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 at all times.
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 list 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 ‘Lucky name numbers’ scenario candidates were asked to:

- Input a person’s name
- Calculate their lucky name number using the grid provided
- Display the names
- Display the lucky name number
- Display the meaning of their lucky name number

Most candidates could create an interface that allowed users to enter their names. The majority of candidates allowed users to enter first name and surname separately.

Many candidates could carry out the first set of calculations and most were able to continue the calculations until the numbers had been resolved to a single digit. Those who arrived at a single digit were able to retrieve the correct definition for the calculated digit.

However, very few candidates had considered the need to validate the user input to ensure that uppercase and lowercase letters could be dealt with and that incorrect input was highlighted by the program.

For the ‘Money Bags’ scenario candidates were asked to:

- Input the weight of a bag of coins
- Input the type of coin
- Calculate how much the weight is different from the correct weight
- For each bag:
  - Display the difference in weight
  - Display number of coins to be added or removed
• When all the bags have been checked;
  o Display total number of bags checked
  o Display total value of the coins added
  o Display total value of the coins removed

Most candidates could provide an interface that allowed the candidates' names to be entered. Many candidates provided facilities that allowed users to enter the type of coins being used and the weight of the bag of coins.

The majority of candidates were able to provide some calculations that identified the difference between the correct weight for a bag of these coins and the weight entered by the user. Very few candidates were able to deal with the final bullet of the scenario.

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.

A significant minority of 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. However, these often did not reflect their original objectives. Many of the evaluation criteria were not expressed in a way that was measurable.

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.

Scenarios are written to ensure that the solutions created will 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 for 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.