



GCSE EXAMINERS' REPORTS

COMPUTER SCIENCE GCSE

SUMMER 2018

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.

Unit	Page
Component 1	1
Component 2	5

COMPUTER SCIENCE

GCSE

Summer 2018

COMPONENT 1: UNDERSTANDING COMPUTER SCIENCE

All Candidates performance across questions						
Question Title	N	Mean	S D	Max Mark	F F	Attempt %
1	2103	5.7	2.3	11	51.4	100.0
2	2095	6.3	4.0	18	34.9	99.6
3	2085	1.8	1.6	7	25.2	99.1
4	2099	1.9	1.8	11	17.2	99.8
5	2017	1.3	1.7	6	22.1	95.9
6	2057	2.2	2.1	10	22.3	97.8
7	1998	1.9	2.0	8	24.3	95.0
8	2058	5.4	2.4	9	60.5	97.9
9	2058	4.4	3.1	8	54.4	97.9
10	2016	1.4	2.3	12	11.3	95.9

As this is the first year of awarding for this specification, it is not possible to compare the general performance of all candidates with previous series. However, it is fair to say from the statistical analyses and Item Level Data that a significant number of candidates found Component 1 challenging, with many candidates failing to meet the demands of a rigorous paper, despite this being of a comparable standard to the Specimen Assessment Materials accredited by the regulator.

It was particularly pleasing, however, to note that the question in which candidates performed best was the question which included significant mathematical content (Q8) and Q9 which required candidates to construct a rather complex truth table for this level.

Generally, however, it was felt by the examining team that candidates seemed under prepared for the demands of this paper and it was slightly disappointing to see so many candidates using a lack of technical detail in responses.

Individual Questions

1. Around half of candidates answered the question on the storage of sound and graphics effectively.

Many candidates were able to correctly identify that the first and third statements about sound sampling were false. However, a minority of candidates were unable to determine that the sound sampling rate statement was also false. This may have been due to a lack of understanding of converting units.

Around half of candidates found it difficult to state what is meant by metadata, but many were able to give a suitable example.

Most candidates were able to explain that lossy compression reduced quality, however fewer could explain that lossless would not increase quality, with some arguing to the

contrary. Very few noted that subsequent lossless compression would increase the file size.

Around half of candidates correctly calculated the file size of the graphic. Many incorrectly identified the graphic as 24KB and did not identify the conversion from bits to bytes.

2. Many candidates identified that the control unit controlled the flow of instructions or data. Only a few candidates, however, identified that it controlled the other parts of the CPU. Most candidates were able to describe the role of the arithmetic logic unit.

A minority of candidates correctly described RISC computers as computers able to complete reduced instruction sets. Only a few went on to explain how this was achieved or its implications.

A very few candidates were able to explain how main memory is used during the fetch-decode-execute cycle. Only a few candidates were able to identify that the instructions to be executed were in main memory. Around half were able to explain that instructions were fetched from 'somewhere' and then decoded and executed. A few were able to explain that data was written back to main memory.

A lack of technical knowledge and use of correct terminology prevented candidates from scoring highly in the question which required a comparison on the performance of two different CPUs. Many candidates were able to correctly identify that CPU 2 has a faster the clock speed and will be able to run faster than CPU 1. However, many failed to refer to the fetch-decode-execute cycle and the processing of more instructions. A majority of candidates were able to identify that CPU 1 has a quad-core whereas CPU 2 is a dual-core CPU. Technical terminology was again the issue, which candidates failing to conclude the impact that this may have in terms of the number of instruction that can be processed by each at the same time. A very few candidates demonstrated an understanding that in theory CPU 1 should be able process instructions twice as fast as CPU 2 and that performance may be affected where one core is waiting on the result of another and therefore cannot carry out any more instructions. Many candidates were able to identify that CPU 2 has more cache memory than CPU 1, but could not conclude how this improves the performance as it can provide instructions and data to the CPU at a much faster rate than other system memory such as RAM. A very few candidates wrote of how CPU 2 will allow more instructions that are repeatedly used by the CPU to be stored.

Many candidates were able to give examples of utility software that could be used to improve disk access speed, but only few could give sufficient examples to gain full marks.

3. A very few candidates were able to describe what is meant by the term embedded systems. The use of an example would have been sufficient, however credit was not given for generic answers such as 'used in a washing machine' as it was not felt that this showed sufficient understanding. A credit worthy example would include 'a system which carries out specific tasks such as a programming cycle of a washing machine'

A very few candidates were able to construct a logic statement to represent the given situation.

Many candidates were able to correctly perform an arithmetic shift left by 3 places the given 16 bit binary number but only a minority were able to state the effect that this has on the number.

4. Only a few candidates were able to correctly identify SMTP as the protocol used to send email messages to an email server.

Very few candidates were able to describe the role of the Ethernet protocol. This was the most poorly answered question in the paper.

Many candidates were able to explain the difference between the HTTP and HTTPS protocols at a simple level in terms of one being secure and the other not, but a very few wrote about the HTTPS protocol working together with the Secure Sockets Layer (SSL) to transport data securely.

A minority of candidates were able to attain one mark for each of the given layers in the TCP/IP 5-layer model. A very few candidates attained two marks.

5. This question was surprisingly poorly answered.

A few candidates were able to describe or give examples of the types of error. Many correlated runtime errors to errors in programming. Many candidates discussed linking errors in terms of hyperlinks and many referred to rounding errors as where a number had been rounded 'incorrectly'.

Where candidates were able to describe the errors, examples generally were provided correctly with the exception of rounding where in most cases an example of rounding was provided but the error not discussed.

6. Most candidates identified that a high level language was closer to English. However, only a few were able to describe other reasons for preferring high level languages.

Very few candidates were able to identify that the interpreter would halt execution and report errors as it encountered them. Fewer still were able to explain that the compiler reported errors in an error log at the end of compilation.

Most candidates were able to identify the difference in speed between compilers and interpreters, however, many answered twice with one being described as faster and the other as slower. A very few were able to identify that interpreters are easier to use as errors can be corrected as they are encountered and fewer still could explain that compilers are more difficult to use as all errors must be corrected after they have been reported in bulk at the end. Many candidates stated that errors are more difficult to find in a compiled program, which is not correct as the error log will clearly identify the location and type of error. A very few identified that compilers produced more efficient object code and that compiled programs did not need to be recompiled.

7. A few candidates identified a record as the most appropriate data structure, however, those that did were in almost all cases able to explain why this was the most suitable.

A few candidates were able to name the types of validation check. In many cases the validation check name had been made up. More were able to design validation checks despite not knowing the correct name for that type of check.

8. As stated previously, this question was reasonably well answered, with many candidates confident in their use of different number counting systems and conversion between them.

For the binary addition question, many candidates answered this well. In a few cases, only 2 marks were accessed for conversion of 67_{10} and 14_{10} into binary, with the addition being incorrect.

A minority of candidates were able to explain the concept of overflow in relation to binary addition by using a suitable example. In some cases it was unclear that the candidate had identified the carry on the MSB. A very few candidates explained that CPU detects that a carry has occurred and sets the overflow flag to true.

9. A majority of candidates were able to correctly complete the given truth table and construct their own. Marks tended to be polarised in this question with candidates gaining 0 marks or full marks.
10. A very few candidates were able to describe the listed forms of attack. However, where candidates did understand the methods they tended to pick up both marks for each.

The extended prose question was very poorly answered by most candidates. Most candidates discussed antivirus, firewall, security through designed and other, sometimes proprietary security software. A very few candidates could briefly discuss ethical hacking. In only a very few cases, candidates displayed good knowledge of ethical hacking, footprinting and penetration testing.

COMPUTER SCIENCE

GCSE

Summer 2018

COMPONENT 2: COMPUTATIONAL THINKING AND PROGRAMMING

General Comments

This was the first year that this was examined. There was an online system for submitting candidate scripts to the WJEC (using SecureAssess) and this largely worked well. Candidates generally prepared the electronic media for delivery to examiners well. However, a small number did have local technical issues. Centres are reminded that they should test that 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.

Good answers were seen for the HTML and Java components, the responses seen for the algorithm, and assembly section were more varied. Those candidates who gained full marks in Java programming question produced very elegant solutions.

Component 2: Comments on individual areas:

Markup Languages

This was answered well by most candidates. However, a significant number of candidates found the link (using `<a href=. . .`) difficult. Candidates also found the `<title>` tag difficult. This was the first time this was assessed and contributed to the difficulty of the paper this year. Some candidates 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.

Algorithms

Some good answers were seen to these question, however, only a few candidates had fully working algorithms. 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.

Object oriented (Java) programming

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 an 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 a differentiator. The code written was generally of a very good standard demonstrating a good understanding of objects, methods and properties within Java. The WJEC emphasises 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 Greenfoot that is defined in the specification. A very small number of centres reported that they had used a different version of Greenfoot in the exam as it had been used during teaching. This is a centre-based issue and should be prevented at all times as it may prevent candidates from accessing marks.



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