



---

# **GCSE MARKING SCHEME**

---

**SUMMER 2019**

**GCSE (NEW)  
COMPUTER SCIENCE - COMPONENT 2  
C500U20-1**

## **INTRODUCTION**

This marking scheme was used by WJEC for the 2019 examination. It was finalised after detailed discussion at examiners' conferences by all the examiners involved in the assessment. The conference was held shortly after the paper was taken so that reference could be made to the full range of candidates' responses, with photocopied scripts forming the basis of discussion. The aim of the conference was to ensure that the marking scheme was interpreted and applied in the same way by all examiners.

It is hoped that this information will be of assistance to centres but it is recognised at the same time that, without the benefit of participation in the examiners' conference, teachers may have different views on certain matters of detail or interpretation.

WJEC regrets that it cannot enter into any discussion or correspondence about this marking scheme.

# WJEC GCSE COMPUTER SCIENCE (NEW)

## SUMMER 2019 MARK SCHEME

### COMPONENT 2 - COMPUTATIONAL THINKING AND PROGRAMMING

#### Guidance for examiners

##### Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers.

These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

#### Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two-stage process.

#### Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance, if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content.

Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

## **Stage 2 – Deciding on the mark**

Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

Q	Answer	Mark	AO1	AO2	AO3	Total
1.	Award <b>one</b> mark for correct HTML Tag		1a			3
(a)	<center>	1				
(b)	<h1>	1				
(c)	<blockquote>	1				
	closing </ . . > Not required, however <> brackets are required.					

Q	Answer	Mark	AO1	AO2	AO3	Total
2.	<p>Award <b>one</b> mark for each correct <u>pair</u> in the correct location: i.e: &lt;h1&gt; &lt;/h1&gt;</p> <p>&lt;a href="url"&gt; &lt;/a&gt; (Note http:// is required or the link will not work correctly on many devices)</p> <p>Accept alternative tags e.g. &lt;big&gt;&lt;/big&gt; instead of &lt;h1&gt;&lt;/h1&gt;, etc</p> <p>Accept alternative HTML (not CSS) solutions which work (only if the identical formatting would be achieved).</p> <pre> &lt;html&gt; &lt;head&gt; &lt;title&gt;  CompSci &lt;/title&gt; &lt;/head&gt;  &lt;body&gt;  &lt;h1&gt; Computer Science! &lt;/h1&gt;  &lt;p&gt; In this technological age, a study of computer science, and particularly how computers are used in the solution of a variety of problems, is essential. &lt;/p&gt;  &lt;img src="eduqas.jpg"&gt;  &lt;p&gt; Learn about:  &lt;ul&gt; &lt;li&gt;Programming&lt;/li&gt; &lt;li&gt;Algorithms&lt;/li&gt; &lt;li&gt;Assembly&lt;/li&gt; &lt;/ul&gt; &lt;/p&gt;  &lt;p&gt; Click the link below to find out more: &lt;/p&gt; &lt;p&gt; &lt;a href= "http://www.eduqas.co.uk/computerscience"&gt;  www.eduqas.co.uk/computerscience  &lt;/a&gt; &lt;/p&gt;  &lt;/body&gt; &lt;/html&gt; </pre> <p>Final correct layout. (Matching the example given in the paper with carriage returns and centre etc)</p>	<p>1 1 1</p> <p>1</p> <p>1</p> <p>1</p> <p>1 ul</p> <p>1 1</p> <p>1</p>		2b		10

Q	Answer	Mark	AO1	AO2	AO3	Total
3.	<p>Indicative content:</p> <p>INP                    'input a number</p> <p>STA first            'store the number a variable called first</p> <p>INP                    'input a number</p> <p>STA second         'store the number in a variable called second</p> <p>LDA first            'load the number in variable first into the accumulator</p> <p>SUB second         'subtract the contents of second from the accumulator</p> <p>OUT                    'output the number in the accumulator</p> <p>first DAT            'declare first as variable</p> <p>second DAT         'declare second as variable</p> <p>Annotation not required.  Order of DAT declared above may be at the beginning or end of the program,  Other programs that provide same output as required by question awarded full credit.</p>	1 1 1 1 1 1 1 1 1 1			3b	9
4.	<p>Award <b>one</b> mark for each correct sentence.</p> <p>(a) Algorithm:  An algorithm is a sequence of instructions(1) that perform a specific task(1).  OR  a process or set of rules to be followed(1) in calculations or other problem-solving operations(1), especially by a computer.</p> <p>(b) Iteration: An iteration is a single pass(1) through a set of instructions. Most programs contain loops of instructions that are executed over and over again. The computer repeatedly executes the loop(1), iterating through the loop.</p> <p>(c) Rogue values are values that <b>fall outside the range or type of the 'normal' data</b> that will be processed by a program.   Accept: A value that will <b>terminate the running</b> of a loop or an algorithm.</p>	1 1  1  1	1b			5

Q	Answer	Mark	AO1	AO2	AO3	Total										
5. (a)	Accept line number <b>or</b> copy of the line. Award <b>one</b> mark for each correct item below.			2b		9										
(i)	Selection = do while (line 11) <b>OR</b> if (line 12)	1														
(ii)	position = position + 1 (line 15)	1														
(iii)	Assignment = line 2 <b>OR</b> line 8 <b>OR</b> line 9 <b>OR</b> line 13 <b>OR</b> line 15	1														
(iv)	Declaration = Declare linearSearch (line 5) <b>OR</b> line 1 <b>OR</b> line 6 <b>OR</b> line 7	1														
(v)	Annotation = {Populate...} (Line 3)	1														
(b)	Award <b>one</b> mark for each correct row completed.															
	<table border="1"> <thead> <tr> <th>Input</th> <th>found =</th> </tr> </thead> <tbody> <tr> <td>linearSearch(dataList, 2)</td> <td>FALSE</td> </tr> <tr> <td>linearSearch(dataList, 3)</td> <td>TRUE</td> </tr> <tr> <td>linearSearch(dataList, 54)</td> <td>FALSE</td> </tr> <tr> <td>linearSearch(dataList, 65)</td> <td>TRUE</td> </tr> </tbody> </table>	Input	found =	linearSearch(dataList, 2)	FALSE	linearSearch(dataList, 3)	TRUE	linearSearch(dataList, 54)	FALSE	linearSearch(dataList, 65)	TRUE	1				
Input	found =															
linearSearch(dataList, 2)	FALSE															
linearSearch(dataList, 3)	TRUE															
linearSearch(dataList, 54)	FALSE															
linearSearch(dataList, 65)	TRUE															
		1														
		1														
		1														





Q	Answer	Mark	AO1	AO2	AO3	Total
7.	Award <b>one</b> mark for each item correctly identified.			2b		5
(a)	<b>World OR Actor</b>	1				
(b)	<b>Ground OR Counter OR Egg OR Meerkat OR Snake</b>	1				
(c)	totalCount	1				
(d)	"cell.jpg" OR 8, 8, 60 OR amount Accept: int amount (not int on its own)	1				
(e)	/** Act */ Ignore carriage returns Accept //	1				
8.	Award <b>one</b> mark per bullet point below:				3b	15
(a)	World is pre-populated on load with: <ul style="list-style-type: none"> <li>one snake only</li> <li>one or more meerkats</li> <li>one or more eggs.</li> </ul>	1 1 1				
(b)	<ul style="list-style-type: none"> <li>meerkats move randomly around world.</li> <li>egg moves randomly around world.</li> <li>random movement implemented using a function (such as getRandomNumber)</li> </ul>	1 1 1				
(c)	<ul style="list-style-type: none"> <li>snake moves around world according to arrow keys.</li> <li>snake moves with appropriate relative speed to egg (equal to or greater than the speed of the egg)</li> </ul>	1 1				
(d)	<ul style="list-style-type: none"> <li>egg is removed from world on collision with snake.</li> </ul>	1				
(e)	<ul style="list-style-type: none"> <li>sound plays when snake and egg collide</li> </ul>	1				
(f)	<ul style="list-style-type: none"> <li>counter added to world.</li> <li>counter increments when snake and egg collide.</li> </ul>	1 1				
(g)	<ul style="list-style-type: none"> <li>counter decrements when meerkat and egg collide.</li> <li>implementation via parameter passing as opposed to wholly new method.</li> </ul>	1 1				
(h)	Greenfoot world saved correctly as finalSnake8	1				