# GCE AS MARKING SCHEME

**SUMMER 2019**

**AS (NEW)**
**COMPUTER SCIENCE - COMPONENT 2**
**B500U20-1**

**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.

# GCE AS COMPUTER SCIENCE

## COMPONENT 2 - PRACTICAL PROGRAMMING TO SOLVE PROBLEMS

## SUMMER 2019 MARK SCHEME

### 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 in **Component 2** the assessment grid advises the marks to allocate to responses which demonstrate the qualities needed in AO2 and AO3. There is limited indicative content as learner response will vary significantly, as the choice of solution will differ based on a variety of factors (e.g. IDE used, interface type chosen, file handling routine used). 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 | Indicative content: | | | All 2.1b | | 12 |

**Person**

| |
|---|
| #firstName : String |
| #surname : String |
| #homeAddress : String |
| #postcode : String |

| |
|---|
| +setFirstName(String) |
| +setSurname(String) |
| +setHomeAddress(String) |
| +setPostcode(String) |

△

**Staff**

| |
|---|
| -staffID : Integer |

| |
|---|
| +setStaffID(Integer) |
| +getStaffID() : Integer |

Award 1 mark **only** for each one of:

| | |
|---|---|
| Superclass Person | 1 |
| Subclass Staff | 1 |
| Correct inheritance order (arrow pointing to Person) | 1 |
| 4 methods in Person | 1 |
| 4 attributes in Person | 1 |
| 2 methods in Staff | 1 |
| 1 attribute in Staff | 1 |
| 4 string parameters in Person | 1 |
| 1 Integer parameter in Staff | 1 |
| 1 Integer returned from Staff method (getStaffID) | 1 |
| all methods public (+) | 1 |
| all attributes private(-) or protected(#) (or mix as above) | 1 |

1

| Q | Answer | Mark | AO1 | AO2 | AO3 | Total |
|---|---|---|---|---|---|---|
| 2 | **Answer must be within the context of IceZone ice rink scenario as all marks are awarded for AO2.1b:**<br><br>**Scenario clearly states the IceZone has non-specialist staff. All answers must reflect this.**<br><br>**Indicative content:**<br><br>IceZone could use a graphical user interface (GUI)<br>A GUI is a type of interface that allows staff to interact with a computer system through graphical icons. | | | all 2.1b | | 8 |

| Advantages | Disadvantages |
|---|---|
| • Intuitive for staff<br>• Easy to navigate for non specialist staff<br>• Uses Windows, Icons, Menus and Pointers which staff may be familiar with<br>• No complicated commands for staff to learn<br>• No need to be a proficient typist | • Requires a large amount of memory so cost for IceZone<br>• Is processor intensive, so cost for IceZone<br>• GUIs take up a much larger amount of hard disk space than other interfaces so cost for IceZone |

IceZone could use a Command Line interface (CLI)

A Command Line Interface is an entirely text-based interface that would allow IceZone staff to communicate with the computer system by typing in commands

| Q | Answer | Mark | AO1 | AO2 | AO3 | Total |
|---|--------|------|-----|-----|-----|-------|
| | <table><tr><th>Advantages</th><th>Disadvantages</th></tr><tr><td>• Quicker to type commands so staff can continue with other tasks<br>• Quicker to input commands as shortcut keys can be used again saves staff time<br>• Little memory and processing power needed compared with other interfaces saving on cost to IceZone<br>• Staff members who have memorised the commands find it efficient to use</td><td>• Confusing for staff who have never used a command line interface<br>• Commands have to be typed precisely. If there is a spelling error the command will fail frustrating non specialist staff members<br>• A large number of commands need to be learned – training costs to IceZone<br>• Not suitable for a novice/difficult for new staff to learn</td></tr></table><br>Award 1 mark for name of UI that IceZone could use x 2<br>Award 1 mark for description of UI that staff could use x 2<br>Award 1 mark for advantage to staff or company of UI x2<br>Award 1 mark for disadvantage to staff or company of UI x2 | | | | | |
| 3 | **Award 1** mark for each correct value | | | | | 8 |

| i: | Position: | Found: | Mark | AO3 |
|----|-----------|--------|------|-----|
| 1 | 2 | FALSE | 1 + 1 | 3.1c |
| 2 | 3 | TRUE | 1 + 1 | 3.1c |
| 3 | 4 | TRUE | 1 + 1 | 3.1c |
| 4 | 5 | TRUE | 1 + 1 | 3.1c |

| Q | Answer | Mark | AO1 | AO2 | AO3 | Total |
|---|--------|------|-----|-----|-----|-------|
| 4 | Any valid/functional loop based algorithm that returns outputs as stated in question:<br><br>**Example**<br><pre>1    currenttemp is real<br>2    set currenttemp = 100<br>3<br>4    begin loop      {start a loop}<br>5<br>6    input currenttemp<br>7<br>8     if currenttemp = 100 then<br>9<br>10  else<br>11<br>12   if currenttemp > -2.1 then<br>13    output "Temperature too high"<br>14   else<br>15<br>16   if currenttemp <= -2.1 AND currenttemp >= -4.0<br>   then<br>17     output "Suitable for general skating"<br>18   else<br>19<br>20   if currenttemp <= -4.1 AND currenttemp >= -5.5<br>   then<br>21     output "Suitable for competition skating"<br>22   else<br>23<br>24  if currenttemp <= -5.6 AND currenttemp >= -9.0<br>   then<br>25    output "Suitable for ice hockey"<br>26   else<br>27<br>28   if currenttemp < -9.0 then<br>29    output "Fault detected"<br>30   end if<br>31<br>32  {Multiple end if may be here}<br>33<br>34 loop until currenttemp = 100<br>35<br>36 End</pre> | | | | 3.1b | 8 |
| | **Award 1** mark for each:<br><br>• Input currenttemp | 1 | | | | |
| | • use of a loop with (100) terminating condition | 1 | | | | |
| | • comparison >-2.1 with output "Temperature too high" | 1 | | | | |
| | • comparison -2.1 to -4.0 with output "Suitable for general skating" | 1 | | | | |
| | • comparison -4.1 to -5.5 with output "Suitable for competition ice skating" | 1 | | | | |
| | • comparison -5.6 to -9.0 with output "Suitable for ice hockey" | 1 | | | | |
| | • comparison < -9.0 with output "Reading error". | 1 | | | | |
| | • Algorithm provides all correct outputs | 1 | | | | |
| | Marks awarded for concepts demonstrated above. Other solutions incorporating above concepts that provide | | | | | |

4

| Q | Answer | Mark | AO1 | AO2 | AO3 | Total |
|---|--------|------|-----|-----|-----|-------|
| | exactly the same result would be awarded credit.<br>Line numbers and indentation not required. | | | | | |
| 5(a)<br>5(b)(i)<br>5(c)(i) | **Indicative content:**<br>• Button available to save data<br>• Data saves<br>• Button available to count staff<br>• The number of staff appears on screen (in any form) | 4 | | | 3.1b | 4 |

| Band | AO3.1b | | |
|------|--------|---|---|
| | **Max 4 marks** | | |
| **3** | **4 marks**<br>The candidate has:<br>• Implemented all the points required as stated in the indicative content<br>• Used and fully exploited the programming facilities of the language<br>• Demonstrated a sound understanding of the appropriate tools and techniques available to them | | |
| **2** | **2-3 marks**<br>The candidate has:<br>• Implemented the majority of the points required as stated in the indicative content. Majority is defined as a response that provides two or three items of the functionality signalled in the indicative content<br>• Used and exploited the programming facilities of the language<br>• Demonstrated an understanding of the tools and techniques available to them | | |
| **1** | **1 mark**<br>The candidate has:<br>• Implemented only one of the points required as stated in the indicative content<br>• Used some of the programming facilities of the language<br>• Demonstrated a limited understanding of the tools and techniques available to them | | |
| **0** | **0 marks**<br>Response not credit worthy or not attempted. | | |

| Q | Answer | Mark | AO1 | AO2 | AO3 | Total |
|---|---|---|---|---|---|---|
| 5(a)(ii)<br>5(b)(ii)<br>5(c)(ii) | **Indicative content:**<br>• Input a number<br>• If the value is 100 the message displayed: "program terminated" program ends.<br>• Positive number or a number above -2.1 is entered the program provides the message "Temperature too high"<br>• Value is between -2.1 to -4.0 the program to provide the message "Suitable for general skating"<br>• Value is between -4.1 to -5.5 the program to provide the message "Suitable for competition ice skating"<br>• Value between -5.6 to -9.0 the program to provide the message "Suitable for ice hockey"<br>• Value is below -9.0 the program to provide the message "Fault detected"<br>• Store the last temperature on disk in a text file called IceResult<br>• Retrieve the last temperature from disk.<br>• Creates a data file called IceResult<br>• Descriptive/useful feedback that file has been saved<br>    ○ Candidates may use custom data types / standard methods / put/get streamwriter/ print # etc<br>• HCI fit for purpose (Textual or GUI) | 12 | | | 3.1b | 12 |

| Band | AO3.1b | | |
|---|---|---|---|
| | **Max 12 marks** | | |
| 3 | **9-12 marks**<br>The candidate has:<br>• Created a new program including all or the majority of the functionality as required in the question and stated in the indicative content. The majority of the functionality is defined as a response that provides nine to twelve items of the functionality signalled in the indicative content<br>• Used and fully exploited the programming facilities of the language<br>• Demonstrated a sound understanding of the appropriate tools and techniques available to them<br>• Written code that is well structured<br>• Provided evidence of a completed user interface which aids user interaction and is intuitive | | |
| 2 | **5-8 marks**<br>The candidate has:<br>• Created a new program including most of the functionality as required in the question and stated in the indicative content. Most of the functionality is defined as a response that provides five to eight items of the functionality signalled in the indicative content<br>• Made use of an appropriate range of the programming facilities of the language<br>• Demonstrated an understanding of the tools and techniques available to them<br>• Provided evidence of a completed user interface which aids user interaction | | |
| 1 | **1-4 marks**<br>The candidate has:<br>• Created a new program with a limited range of the functionality as stated in the indicative content or improved the prototype provided by adding a limited range of the new functionality as stated in the indicative content. A limited range of functionality is defined as a response that provides one to four items of the functionality signalled in the indicative content<br>• Used a limited range of the programming facilities of the language<br>• Demonstrated a limited understanding of the tools and techniques available to them<br>• Provided evidence of a user interface | | |
| 0 | **0 marks**<br>Response not credit worthy or not attempted. | | |

| Q | Answer | Mark | AO1 | AO2 | AO3 | Total |
|---|---|---|---|---|---|---|
| 5(a)(iii)<br>5(b)(iii)<br>5(c)(iii) | **Indicative content:**<br>Clear annotation of steps within the following routines:<br>• The logic of one of the above -2.1 statement.<br>• The -2.1 to -4 logic<br>• The -4.1 to -5.5 logic<br>• The -5.8 to -9 logic<br>• The below -9 logic<br>• The terminating condition logic<br>• Storage of data to file or Retrieving data from file<br>• Use of self-documenting identifiers / explanation of variables | 8 | | | 3.1a | 8 |

| Band | AO3.1a |
|---|---|
| | **Max 8 marks** |
| 3 | **7-8 marks**<br>The candidate has:<br>• Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **all** programming routines listed in the indicative content<br>• Written code using self-documenting identifiers / explained variables<br>• Used appropriate technical terminology referring to the indicative content confidently and accurately. |
| 2 | **3-6 marks**<br>**5-6** marks can be awarded if the candidate has:<br>• Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **all** programming routines listed in the indicative content<br>• Not written code using self-documenting identifiers / not explained variables<br>• Used appropriate technical terminology referring to the indicative content.<br>OR<br>• Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **4-5** of the programming routines listed in the indicative content<br>• Written code using self-documenting identifiers / explained variables<br>• Used appropriate technical terminology referring to the indicative content.<br><br>**3-4** marks can be awarded if the candidate has:<br>• Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **3-4** of the programming routines listed in the indicative content<br>• Not written code using self-documenting identifiers / not explained variables<br>• Used appropriate technical terminology referring to the indicative content.<br>OR<br>• Produced listings that are appropriately laid out and included sufficient annotation to demonstrate an understanding of **one** of the programming routines listed in the indicative content<br>• Written code using self-documenting identifiers / explained variables<br>• Used appropriate technical terminology referring to the indicative content. |
| 1 | **1-2 marks**<br>The candidate has:<br>• Produced listings that are appropriately laid out and include sufficient annotation to demonstrate an understanding of **one** programming routine listed in the indicative content<br>• Used limited technical terminology referring to the indicative content.<br>OR<br>• Written code using self-documenting identifiers<br>• Used limited technical terminology referring to the indicative content. |
| 0 | **0 marks**<br>Response not credit worthy or not attempted. |