



Cambridge O Level

COMPUTER SCIENCE

2210/22

Paper 2

October/November 2021

MARK SCHEME

Maximum Mark: 50

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the October/November 2021 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

This document consists of **13** printed pages.

PUBLISHED**Generic Marking Principles**

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

GENERIC MARKING PRINCIPLE 6:

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

PUBLISHED

| Question | Answer | Marks | | | | | | | | | | | | |
|------------------|--|-------------|--|-------------|-----|---------------|--------|----|---|-------------|------|------|--|----------|
| Section A | | | | | | | | | | | | | | |
| 1(a)(i) | <p>One mark per point</p> <ul style="list-style-type: none"> • Variable PassengerID// StartStage • Use Storing the unique ID number of the passenger// Storing/inputting the start stage of the journey | 2 | | | | | | | | | | | | |
| 1(a)(ii) | <p>One mark per point</p> <p>MP1 Name of array MP2 Data type of array MP3 Sample data for array MP4 Use of array MP5 At least two complete arrays with all of the above</p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">Array name</th> <th style="text-align: left;">Data type</th> <th style="text-align: left;">Sample data</th> <th style="text-align: left;">Use</th> </tr> </thead> <tbody> <tr> <td>JourneyStage1</td> <td>string</td> <td>C1</td> <td>to store the code for the home to start station</td> </tr> <tr> <td>PriceStage1</td> <td>real</td> <td>1.50</td> <td>to store the price of first stage of the journey</td> </tr> </tbody> </table> | Array name | Data type | Sample data | Use | JourneyStage1 | string | C1 | to store the code for the home to start station | PriceStage1 | real | 1.50 | to store the price of first stage of the journey | 5 |
| Array name | Data type | Sample data | Use | | | | | | | | | | | |
| JourneyStage1 | string | C1 | to store the code for the home to start station | | | | | | | | | | | |
| PriceStage1 | real | 1.50 | to store the price of first stage of the journey | | | | | | | | | | | |
| 1(b) | <p>One mark per bullet point</p> <p>MP1 Use of validation check, e.g. range check, type check, presence check, length check, format check MP2 Use of conditional statement to check if the validation fails ... MP3 ... a re-entry is requested MP4 Use of loop to repeat the process until an acceptable answer is input MP5 More than one appropriate validation check used / described.</p> | 3 | | | | | | | | | | | | |
| 1(c) | <p>Any six from:</p> <p>MP1 Conditional statement to check departure time against 10:00 MP2 ... calculate 40% discount // calculate 60% of the original price MP3 ... calculate discounted total price MP4 Output the discounted total price MP5 Output the booking details with suitable messages MP6 Input with prompt for passenger confirmation ... MP7 ... attempt at action following the confirmation input MP8 Repeating booking data entry if incorrect MP9 Re-checking journey details for correctness</p> | 6 | | | | | | | | | | | | |

| Question | Answer | Marks |
|----------|---|----------|
| 1(c) | <p>Example answer</p> <pre>// Tasks 1 and 2 completed IF CollectedTime[Index] > 10:00 THEN JourneyCost[Index] ← JourneyCost[Index] * 0.6 ENDF PRINT "Your journey cost is: ", JourneyCost[Index] PRINT "Your journey details are: ", PassengerID[Index], JourneyTime[Index], JourneyCodes[Index], JourneyID[Index] PRINT "Are these details correct? (Y or N)" INPUT Correct IF Correct = "N" THEN WHILE Correct = "N" PRINT "Re-enter your journey details" PRINT "Correct passenger ID " INPUT PassengerID[Index] PRINT "Correct journey time " INPUT JourneyTime[Index] PRINT "Correct journey codes " INPUT JourneyCodes[Index] PRINT "Your revised journey details are: ", PassengerID[Index], JourneyTime[Index], JourneyCodes[Index] PRINT "Are these details correct? (Y or N)" INPUT Correct ENDWHILE ENDF //Program continues</pre> | |
| 1(d) | <p>Explanation of how each of the following could be done</p> <p>Any four from:</p> <p>MP1 Declaring/using a counter to store the number of bookings for each passenger</p> <p>MP2 Updating the counter for the number of bookings made by each passenger</p> <p>MP3 Attempt to check the number of bookings ...</p> <p>MP4 ... for the correct condition e.g. if the number of bookings is more than 10 / equal to 10</p> <p>MP5 Apply the extra discount to the total price of future journeys</p> | 4 |

PUBLISHED

| Question | Answer | Marks | | | | | | | | | | | | | | | | | | | | |
|--|---|---------------------|-------------------|---------------------|----------------|--|--|---|--|---|---|--|--|---|---|--|--|--|--|--|---|----------|
| | Section B | | | | | | | | | | | | | | | | | | | | | |
| 2 | <p>One mark for two correct rows Two marks for three correct rows Three marks for four correct rows.</p> <table border="1"> <thead> <tr> <th>Statement</th> <th>Validation (✓)</th> <th>Verification (✓)</th> <th>Neither (✓)</th> </tr> </thead> <tbody> <tr> <td>a check where data is re-entered to make sure no errors have been introduced during data entry</td> <td></td> <td>✓</td> <td></td> </tr> <tr> <td>an automatic check to make sure the data entered has the correct number of characters</td> <td>✓</td> <td></td> <td></td> </tr> <tr> <td>a check to make sure the data entered is sensible</td> <td>✓</td> <td></td> <td></td> </tr> <tr> <td>a check to make sure the data entered is correct</td> <td></td> <td></td> <td>✓</td> </tr> </tbody> </table> | Statement | Validation (✓) | Verification (✓) | Neither (✓) | a check where data is re-entered to make sure no errors have been introduced during data entry | | ✓ | | an automatic check to make sure the data entered has the correct number of characters | ✓ | | | a check to make sure the data entered is sensible | ✓ | | | a check to make sure the data entered is correct | | | ✓ | 3 |
| Statement | Validation (✓) | Verification (✓) | Neither (✓) | | | | | | | | | | | | | | | | | | | |
| a check where data is re-entered to make sure no errors have been introduced during data entry | | ✓ | | | | | | | | | | | | | | | | | | | | |
| an automatic check to make sure the data entered has the correct number of characters | ✓ | | | | | | | | | | | | | | | | | | | | | |
| a check to make sure the data entered is sensible | ✓ | | | | | | | | | | | | | | | | | | | | | |
| a check to make sure the data entered is correct | | | ✓ | | | | | | | | | | | | | | | | | | | |

| Question | Answer | Marks |
|----------|---|----------|
| 3 | <p>One mark per bullet point</p> <p>Normal test data</p> <ul style="list-style-type: none"> • Test data e.g. 50 (allow any number between 1 and 100 inclusive) • Reason Data that is within range and should be accepted <p>Extreme test data</p> <ul style="list-style-type: none"> • Test data 100 / 1 • Reason Data at the maximum / minimum end of the range and should be accepted <p>Erroneous test data</p> <ul style="list-style-type: none"> • Test data e.g. 300 (allow anything that isn't between 1 and 100 inclusive, including other data types) • Reason Data outside the range that should be rejected | 6 |

PUBLISHED

| Question | Answer | Marks |
|----------|---|----------|
| 4(a) | <p>One mark for error identified and suggested correction (Max three)</p> <p>Line 8 OUTPUT Value2 – should be INPUT Value2 Line 9 IF Operator – should be CASE OF Operator Line 15 OUTPUT "The answer is ", Value1 – should be Answer</p> <p>The loop may be corrected using a number of alternative methods:</p> <p>One mark for error identified and suggested correction (Max two)</p> <p>Method 1 Line 1 Continue ← 1 should be Continue ← 0 Line 22 UNTIL Continue = 0 should be ENDWHILE // Line 2 WHILE Continue = 0 should be REPEAT and Line 22 UNTIL Continue = 0 should be Until Continue = 1</p> <p>OR</p> <p>Method 2 Line 2 WHILE Continue = 0 should be REPEAT Line 20 Continue ← 1 should be Continue ← 0 // Line 1 Continue ← 1 should be Continue ← 0 and Line 22 UNTIL Continue = 0 should be Until Continue = 1</p> <p>OR</p> <p>Method 3 Line 2 WHILE Continue = 0 should be WHILE Continue = 1 Line 20 Continue ← 1 should be Continue ← 0 and Line 22 UNTIL Continue = 0 should be ENDWHILE</p> | 5 |

| Question | Answer | Marks |
|----------|---|-------|
| 4(a) | <p>Corrected algorithm example 1</p> <pre> 1 Continue ← 0 2 WHILE Continue = 0 (DO) 3 OUTPUT "Enter 1 for +, 2 for -, 3 for * or 4 for /" 4 INPUT Operator 5 OUTPUT "Enter the first value" 6 INPUT Value1 7 OUTPUT "Enter the second value" 8 INPUT Value2 9 CASE OF Operator 10 1: Answer ← Value1 + Value2 11 2: Answer ← Value1 - Value2 12 3: Answer ← Value1 * Value2 13 4: Answer ← Value1 / Value2 14 ENDCASE 15 OUTPUT "The answer is ", Answer 16 OUTPUT "Do you wish to enter more values (Yes or No)?" 17 INPUT MoreValues 18 IF MoreValues = "No" 19 THEN 20 Continue ← 1 21 ENDIF 22 ENDWHILE </pre> | |

| Question | Answer | Marks |
|----------|--|-------|
| 4(a) | <p>Corrected algorithm example 2</p> <pre> 1 Continue ← 1 2 REPEAT 3 OUTPUT "Enter 1 for +, 2 for -, 3 for * or 4 for /" 4 INPUT Operator 5 OUTPUT "Enter the first value" 6 INPUT Value1 7 OUTPUT "Enter the second value" 8 INPUT Value2 9 CASE OF Operator 10 1: Answer ← Value1 + Value2 11 2: Answer ← Value1 - Value2 12 3: Answer ← Value1 * Value2 13 4: Answer ← Value1 / Value2 14 ENDCASE 15 OUTPUT "The answer is ", Answer 16 OUTPUT "Do you wish to enter more values (Yes or No)?" 17 INPUT MoreValues 18 IF MoreValues = "No" 19 THEN 20 Continue ← 0 21 ENDIF 22 UNTIL Continue = 0 </pre> | |

PUBLISHED

| Question | Answer | Marks |
|----------|---|----------|
| 4(b) | <p>One mark per bullet</p> <p>MP1 Appropriate loop (begin and end) / otherwise selection</p> <p>MP2 Testing both ends of condition</p> <p>MP3 Suitable message</p> <p>MP4 Input/re-input</p> <pre> WHILE Operator < 1 OR Operator > 4 (DO) OUTPUT "Enter 1, 2, 3 or 4" INPUT Operator ENDWHILE Alternative answer REPEAT IF Operator < 1 OR Operator > 4 THEN OUTPUT "Enter 1, 2, 3 or 4" INPUT Operator ENDIF UNTIL Operator >= 1 AND Operator <= 4 </pre> <p>One mark</p> <p>After line 4 / between lines 2 and 5</p> | 5 |

| Question | Answer | | | | Marks | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|--|-------|-------|--------|-------|-------|-------|-------|--------|--|--|---|---|--|---|--|--|--|--|--|----|--|----|--|---|--|--|--|--|--|----|--|----|--|---|--|--|--|--|--|----|----|--|--|---|--|--|--|--|--|----|--|----|--|---|--|--|--|--|---|
| 5 | <p>One mark for each correct column</p> <table border="1" data-bbox="338 284 1319 1003"> <thead> <tr> <th data-bbox="338 284 486 347">List</th> <th data-bbox="486 284 631 347">Value</th> <th data-bbox="631 284 779 347">List1</th> <th data-bbox="779 284 925 347">List2</th> <th data-bbox="925 284 1319 347">OUTPUT</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>77</td> <td></td> <td>77</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>16</td> <td></td> <td>93</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>35</td> <td>35</td> <td></td> <td></td> </tr> <tr> <td>2</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>-7</td> <td></td> <td>86</td> <td></td> </tr> <tr> <td>5</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> | | | | List | Value | List1 | List2 | OUTPUT | | | 0 | 0 | | 2 | | | | | | 77 | | 77 | | 2 | | | | | | 16 | | 93 | | 1 | | | | | | 35 | 35 | | | 2 | | | | | | -7 | | 86 | | 5 | | | | | 5 |
| List | Value | List1 | List2 | OUTPUT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 77 | | 77 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 16 | | 93 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 35 | 35 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | -7 | | 86 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| Question | Answer | | | | | Marks |
|----------|-------------|--------------|--------------|--------------|--------------------|-------|
| 5 | List | Value | List1 | List2 | OUTPUT | |
| | | 18 | | | Input Error | |
| | 1 | | | | | |
| | | 11 | 46 | | | |
| | 1 | | | | | |
| | | 12 | 58 | | | |
| | 2 | | | | | |
| | | 20 | | 106 | | |
| | -1 | | | | List 1 = 58 | |
| | | | | | List 2 = 106 | |
| | | | | | List 2 is greatest | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| Question | Answer | | | | | Marks | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|-------------------------------------|--------------------------|--------------------------|--------------------------|--------|-----------|-------------|--------|---------|--|--------|-------|-------|-------|-------|--|-------|-----------|--|--|--|--|-------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-----------|--|--|--------|------|--|-----|--|--|--|--|--|---|
| 6(a)(i) | InStock | | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6(a)(ii) | ProductID | | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6(b) | <p>One mark for correct fieldnames One mark for correct table names and show fields One mark for correct sort One mark for correct search criteria in all columns</p> <table border="1" data-bbox="488 517 1621 906"> <tr> <td data-bbox="389 517 488 580">Field:</td> <td data-bbox="488 517 712 580">ProductID</td> <td data-bbox="712 517 943 580">ProductName</td> <td data-bbox="943 517 1167 580">Animal</td> <td data-bbox="1167 517 1391 580">InStock</td> <td data-bbox="1391 517 1621 580"></td> </tr> <tr> <td data-bbox="389 580 488 644">Table:</td> <td data-bbox="488 580 712 644">STOCK</td> <td data-bbox="712 580 943 644">STOCK</td> <td data-bbox="943 580 1167 644">STOCK</td> <td data-bbox="1167 580 1391 644">STOCK</td> <td data-bbox="1391 580 1621 644"></td> </tr> <tr> <td data-bbox="389 644 488 708">Sort:</td> <td colspan="5" data-bbox="488 644 1621 708">Ascending</td> </tr> <tr> <td data-bbox="389 708 488 772">Show:</td> <td data-bbox="488 708 712 772"><input checked="" type="checkbox"/></td> <td data-bbox="712 708 943 772"><input checked="" type="checkbox"/></td> <td data-bbox="943 708 1167 772"><input type="checkbox"/></td> <td data-bbox="1167 708 1391 772"><input type="checkbox"/></td> <td data-bbox="1391 708 1621 772"><input type="checkbox"/></td> </tr> <tr> <td data-bbox="389 772 488 836">Criteria:</td> <td colspan="2" data-bbox="488 772 943 836"></td> <td data-bbox="943 772 1167 836">="cat"</td> <td data-bbox="1167 772 1391 836">=Yes</td> <td data-bbox="1391 772 1621 836"></td> </tr> <tr> <td data-bbox="389 836 488 900">or:</td> <td colspan="5" data-bbox="488 836 1621 900"></td> </tr> </table> | | | | | Field: | ProductID | ProductName | Animal | InStock | | Table: | STOCK | STOCK | STOCK | STOCK | | Sort: | Ascending | | | | | Show: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Criteria: | | | ="cat" | =Yes | | or: | | | | | | 4 |
| Field: | ProductID | ProductName | Animal | InStock | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Table: | STOCK | STOCK | STOCK | STOCK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Sort: | Ascending | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Show: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Criteria: | | | ="cat" | =Yes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| or: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |