



# Cambridge O Level

---

**COMPUTER SCIENCE**

**2210/23**

Paper 2 Problem-solving and Programming

**October/November 2022**

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 2022 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

---

This document consists of **10** 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.

**Please note the following further points:**

The words in **bold** in the mark scheme are important text that needs to be present, or some notion of it needs to be present. It does not have to be the exact word, but something close to the meaning.

If a word is underlined, this **exact** word must be present.

A single forward slash means this is an alternative word. A double forward slash means that this is an alternative mark point.

Ellipsis (...) on the end of one-mark point and the start of the next means that the candidate **cannot** get the second mark point without being awarded the first one. If a mark point has an ellipsis at the beginning, but there is no ellipsis on the mark point before it, then this is just a follow-on sentence and **can** be awarded **without** the previous mark point.

**PUBLISHED**

Question	Answer	Marks
<b>Section A</b>		
1(a)(i)	<p>Many correct answers, the name used must be meaningful. The name given is an example only.</p> <p><b>One</b> mark per mark point, max <b>three</b></p> <p>Constant name     <code>BasicClean</code>  Value                <code>10.00</code>  Reason                this number will not change whilst the program is running</p>	<b>3</b>
1(a)(ii)	<p>Description  Many correct answers, the name used must be meaningful. The name is an example only.</p> <p><b>One</b> mark per mark point, max <b>three</b></p> <p>MP1     name e.g. <code>NameAddress</code>  MP2     appropriate data type e.g. <code>string</code>  MP3     sample data for the array e.g. <code>Mrs Singh, Park View</code></p>	<b>3</b>
1(b)	<p><b>One</b> mark per mark point, max <b>three</b></p> <p>MP1     use a conditional statement // an example of conditional statement used //  use an input statement with validation// an example of input statement used  MP2     any working method – for example, to only store <code>True</code> in the three floors array if the two floors array has not been set to <code>True</code>  MP3     output an appropriate error message when an error condition has occurred //  choose appropriate routes through the program depending upon the choice of floors</p>	<b>3</b>

Question	Answer	Marks
1(c)	<p><b>One mark per mark point, max six</b></p> <p>MP1 initialise bill total to 10  MP1 check for additional windows  MP2 ... and add in amount to total if required  MP3 check for 2 floors <b>or</b> 3 floors  MP4 ... multiply by 1.1 <b>or</b> 1.15 as required  MP5 check for inside and multiply by 1.25 if required  MP6 check for polish and multiply by 1.05 if required  MP7 MP2 to MP7 completed in correct order  MP8 check for solar panels and add 20.00 if required</p> <p><b>Sample</b></p> <pre>Total ← 10.00 ExtraWindowsCost ← INT((NumberExtraWindows[Bill] -1) / 5) * 5.00 Total ← Total + ExtraWindowsCost CASE NumberOfFloors[Bill] OF // check number of floors   2 : Total ← Total * 1.1   3 : Total ← Total * 1.15 ENDCASE IF Inside[Bill] // check for cleaning inside windows   THEN     Total ← Total * 1.25   ENDF IF Polish[Bill] // check for polishing windows   THEN     Total ← Total * 1.05   ENDF IF Solar[Bill] // check for cleaning solar panels   THEN     Total ← Total + 20.00   ENDF</pre>	6

Question	Answer	Marks
1(d)	<p>Explanation</p> <p><b>One</b> mark per mark point, max <b>five</b></p> <p>MP1    how the program counted the number of customers using a service</p> <p>MP1    ... how the program counted this for all services</p> <p>MP2    ... excluding basic windows and additional windows</p> <p>MP3    how the program identified the service with the largest number of customers</p> <p>MP4    how the program identified the service with the smallest number of customers</p> <p>MP5    how the program calculated a percentage of the total number of bills stored...</p> <p>MP6    ... for largest and smallest</p> <p>MP7    how the program displayed these percentages with suitable messages</p> <p>Programming statements when used must be explained.</p>	<b>5</b>

Question	Answer	Marks
<b>Section B</b>		
2(a)	<p><b>One mark per mark point in the correct position, max six</b></p> <ul style="list-style-type: none"> <li>• line 01 50</li> <li>• line 08 Value &gt; 50</li> <li>• line 12 Reading[Value] + 1</li> <li>• line 18 INPUT Value</li> <li>• line 23 Reading[Count]</li> <li>• line 24 Count + 1</li> </ul>	<b>6</b>
2(b)	<p><b>One mark per place in code and action, max three</b></p> <ul style="list-style-type: none"> <li>• line 21 set Count to 50 / Count ← 50</li> <li>• line 24 subtract 1 from Count / Count ← Count -1</li> <li>• line 25 check for Count equal to 34 / check for Count less than 35 / UNTIL Count = 34 / UNTIL Count &lt; 35</li> </ul> <p>or</p> <p><b>One mark per place in code and action, max three</b></p> <ul style="list-style-type: none"> <li>• line 21 set up FOR loop stating at 50 and finishing at 35 / FOR Count ← 50 TO 35 STEP -1</li> <li>• Remove lines 22 and 24</li> <li>• line 25 End FOR loop / NEXT Count</li> </ul> <p><b>Examples</b></p> <pre>21 Count ← 50 22 REPEAT 23 OUTPUT "There are ", Reading[Count], " readings, " of ", Count 24 Count ← Count -1 25 UNTIL Count = 34</pre> <pre>21 FOR Count ← 50 TO 35 STEP -1 23 OUTPUT "There are ", Reading[Count], " readings, " of ", Count 25 NEXT Count</pre>	<b>3</b>

Question	Answer	Marks
3(a)(i)	<ul style="list-style-type: none"> <li>• 1</li> </ul>	<b>1</b>
3(a)(ii)	<p><b>One</b> mark for each correct letter seen, max <b>two</b></p> <ul style="list-style-type: none"> <li>• C</li> <li>• D</li> </ul>	<b>2</b>
3(b)(i)	<p><b>One</b> mark per mark point, max <b>two</b></p> <ul style="list-style-type: none"> <li>• two or more digits</li> <li>• transposed</li> </ul>	<b>2</b>
3(b)(ii)	<p><b>One</b> mark per mark point, max <b>two</b></p> <ul style="list-style-type: none"> <li>• multiply each digit by a different number / its place value</li> <li>• before adding them together and dividing by a number</li> </ul>	<b>2</b>
3(c)	<p><b>One</b> mark per mark point, max <b>two</b></p> <ul style="list-style-type: none"> <li>• length check</li> <li>• type check</li> <li>• presence check</li> <li>• format check</li> </ul>	<b>2</b>



Question	Answer	Marks																																												
4(a)	<p><b>One mark for each correct column, max four</b></p> <table border="1" data-bbox="696 288 1581 1007"> <thead> <tr> <th>Stock</th> <th>Total</th> <th>Sale</th> <th>OUTPUT</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>0</td> <td></td> <td></td> </tr> <tr> <td>9</td> <td>1</td> <td>Y</td> <td></td> </tr> <tr> <td>8</td> <td>2</td> <td>Y</td> <td></td> </tr> <tr> <td>7</td> <td>3</td> <td>Y</td> <td></td> </tr> <tr> <td>6</td> <td>4</td> <td>Y</td> <td></td> </tr> <tr> <td>5</td> <td>5</td> <td>Y</td> <td></td> </tr> <tr> <td>4</td> <td>6</td> <td>Y</td> <td></td> </tr> <tr> <td>14</td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td>N</td> <td>Add new stock</td> </tr> <tr> <td></td> <td></td> <td></td> <td>6</td> </tr> </tbody> </table>	Stock	Total	Sale	OUTPUT	10	0			9	1	Y		8	2	Y		7	3	Y		6	4	Y		5	5	Y		4	6	Y		14						N	Add new stock				6	<b>4</b>
Stock	Total	Sale	OUTPUT																																											
10	0																																													
9	1	Y																																												
8	2	Y																																												
7	3	Y																																												
6	4	Y																																												
5	5	Y																																												
4	6	Y																																												
14																																														
		N	Add new stock																																											
			6																																											
4(b)	<p><b>One mark per mark point, max three</b></p> <ul style="list-style-type: none"> <li>• input a number / quantity</li> <li>• ... check that the number is less than (or equal to) the number in stock</li> <li>• ... after checking for N</li> <li>• update Stock by that number / <math>Stock \leftarrow Stock - Number</math></li> <li>• update Total by that number / <math>Total \leftarrow Total + Number</math></li> </ul>	<b>3</b>																																												

Question	Answer	Marks																														
5(a)	<p><b>One</b> mark per mark point, max <b>two</b></p> <ul style="list-style-type: none"> <li>• ItemCode</li> <li>• uniquely identifies each item</li> </ul>	<b>2</b>																														
5(b)	<p><b>One</b> mark per mark point, max <b>three</b></p> <ul style="list-style-type: none"> <li>• correct rows Field, Table and Sort</li> <li>• correct row Show</li> <li>• correct Criteria row</li> </ul> <div style="margin-left: 100px;"> <table border="1" style="border-collapse: collapse; width: 250px;"> <tr> <td style="padding: 2px;">Field:</td> <td style="padding: 2px;">Description</td> <td style="padding: 2px;">Country</td> <td style="padding: 2px;">InStore</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">Table:</td> <td style="padding: 2px;">MUSEUM</td> <td style="padding: 2px;">MUSEUM</td> <td style="padding: 2px;">MUSEUM</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">Sort:</td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">Show:</td> <td style="padding: 2px; text-align: center;"><input checked="" type="checkbox"/></td> <td style="padding: 2px; text-align: center;"><input checked="" type="checkbox"/></td> <td style="padding: 2px; text-align: center;"><input type="checkbox"/></td> <td style="padding: 2px; text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td style="padding: 2px;">Criteria:</td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px; text-align: center;">N</td> <td style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">or:</td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> <td style="padding: 2px;"></td> </tr> </table> </div>	Field:	Description	Country	InStore		Table:	MUSEUM	MUSEUM	MUSEUM		Sort:					Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Criteria:			N		or:					<b>3</b>
Field:	Description	Country	InStore																													
Table:	MUSEUM	MUSEUM	MUSEUM																													
Sort:																																
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																												
Criteria:			N																													
or:																																