Cambridge Assessment International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/23**

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.

© UCLES 2022

[Turn over

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.

© UCLES 2022

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.

© UCLES 2022

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

© UCLES 2022

Question	Answer	Marks

1(c)	**One** mark per mark point, max **six**
	
	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 **or** 3 floors
	MP4 … multiply by 1.1 **or** 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
	
	Sample
	`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`
	`ENDIF`
	`IF Polish[Bill] // check for polishing windows`
	`   THEN`
	`    Total ← Total * 1.05`
	`ENDIF`
	`IF Solar[Bill] // check for cleaning solar panels`
	`  THEN`
	`    Total ← Total + 20.00`
	`ENDIF`
		6

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

© UCLES 2022

Question	Answer	Marks
	Section B	
2(a)	**One** mark per mark point in the correct position, max **six** • line 01 `50` • line 08 `Value > 50` • line 12 `Reading[Value] + 1` • line 18 `INPUT Value` • line 23 `Reading[Count]` • line 24 `Count + 1`	**6**

2(b)	**One** mark per place in code **and** action, max **three**
	
	• line 21 set Count to 50 / `Count ← 50`
	• line 24 subtract 1 from Count / `Count ← Count -1`
	• line 25 check for Count equal to 34 / check for Count less than 35 / `UNTIL Count = 34` / `UNTIL Count < 35`
	or
	One mark per place in code **and** action, max **three**
	
	• line 21 set up FOR loop stating at 50 and finishing at 35 / `FOR Count ← 50 TO 35 STEP -1`
	• Remove lines 22 and 24
	• line 25 End FOR loop / `NEXT Count`
	
	Examples
	`21 Count ← 50`
	`22 REPEAT`
	`23 OUTPUT "There are ", Reading[Count], " readings, " of ", Count`
	`24 Count ← Count -1`
	`25 UNTIL Count = 34`
	
	`21 FOR Count ← 50 TO 35 STEP -1`
	`23 OUTPUT "There are ", Reading[Count], " readings, " of ", Count`
	`25 NEXT Count`
		3

© UCLES 2022

Question	Answer	Marks
3(a)(i)	• 1	**1**
3(a)(ii)	**One** mark for each correct letter seen, max **two** • C • D	**2**
3(b)(i)	**One** mark per mark point, max **two** • two or more digits • transposed	**2**
3(b)(ii)	**One** mark per mark point, max **two** • multiply each digit by a different number / its place value • before adding them together and dividing by a number	**2**
3(c)	**One** mark per mark point, max **two** • length check • type check • presence check • format check	**2**

© UCLES 2022

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4(a)</td><td>One mark for each correct column, max four
<table>
<tr><th>Stock</th><th>Total</th><th>Sale</th><th>OUTPUT</th></tr>
<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>
</table></td><td>4</td></tr>
<tr><td>4(b)</td><td>One mark per mark point, max three

• input a number / quantity
• … check that the number is less than (or equal to) the number in stock
• … after checking for N
• update <code>Stock</code> by that number / <code>Stock ← Stock – Number</code>
• update <code>Total</code> by that number / <code>Total ← Total + Number</code></td><td>3</td></tr>
</table>

© UCLES 2022

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>One mark per mark point, max two

• ItemCode
• uniquely identifies each item</td><td>2</td></tr>
<tr><td>5(b)</td><td>One mark per mark point, max three

• correct rows Field, Table and Sort
• correct row Show
• correct Criteria row

<table>
<tr><td>Field:</td><td>Description</td><td>Country</td><td>InStore</td><td></td></tr>
<tr><td>Table:</td><td>MUSEUM</td><td>MUSEUM</td><td>MUSEUM</td><td></td></tr>
<tr><td>Sort:</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Show:</td><td>☑</td><td>☑</td><td>☐</td><td>☐</td></tr>
<tr><td>Criteria:</td><td></td><td></td><td>N</td><td></td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td></td></tr>
</table>
</td><td>3</td></tr>
</table>

© UCLES 2022