Cambridge Assessment
International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/22**

Paper 2 **March 2020**

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 March 2020 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

This document consists of **9** printed pages.

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.

Question	Answer	Marks
	Section A	
1(a)	Any appropriate constant, for example: **Constant name** `noTradeInDiscount` **Value** 0.05/5 **Use** Storing the percentage discount if no car is traded in	**3**
1(b)	**Variable name** `TradeInValue` // any suitable variable name for trade-in amount **Data type** real/integer **Validation** any **two** from • Use of conditional /loop statement / `IF … THEN` / `REPEAT … UNTIL` / `WHILE` • value input is greater than (or equal to) 10 000/0.1 / less than (or equal to) 100 000/1 / numeric // a value is actually input // error message for rejected input • range check // type check // presence check	**4**
1(c)	Any **four** from: MP1 Display extra model/choice MP2 Increase options available to 4 MP3 Extend validation rule / selection statement(s) for input selection MP4 Store name of extra model e.g. new variable/constant/extra item in array MP5 Store price of extra model e.g. new variable/ constant/extra item in array MP6 Change code to calculate price to pay to include extra model	**4**

Question	Answer	Marks

1(d)	Any **five** from:
	MP1 Calculate cashback for payment method 1
	MP2 Calculate monthly payments for payment method 2
	MP3 Calculate amount and monthly payments for payment method 3
	MP4 Method 1 – output full amount, one payment and cashback
	MP5 Method 2 – output full amount, full amount divided by 48 and 48 payments
	MP6 Method 3 – output full amount times 1.05, full amount times 1.05 divided by 84 and 84 payments
	MP7 Attempt to output showing all 3 payment methods with appropriate messages (MPs 4,5,6 not awarded)
	
	Sample answer:
	`PRINT "Full Amount ",amountToPay, " to pay.  Number of Payments is 1"`
	`PRINT "Cashback ", amountToPay * 0.01`
	`PRINT "Over four years ",amountToPay, " to pay.  Number of Payments is 48"`
	`PRINT "Each monthly payment is ", amountToPay / 48`
	`PRINT "Over seven years ",amountToPay * 1.05, " to pay.  Number of Payments is 84"`
	`PRINT "Each monthly payment is ", amountToPay * 1.05 / 84`
		5

1(e)	Explanation Any **four** from: MP1 Use of selection statement to check if customer chose to pay the full amount MP2 Consideration of special case where there are no optional extras chosen MP3 Comparison of 1% of total price to pay with the total cost of optional extras MP4 Use of selection statement to check for the largest value of cashback or extras /smallest amount to pay MP5 … display the cost of this option first MP6 … then display the cost of the other option	**4**

Question	Answer	Marks
Section B		
2(a)	Error: • Problem with zero … • … stored in the negative number array // negative number count increases by 1 Correction: • Replace ELSE with IF • `IF Number < 0 (THEN)`	**4**
2(b)	Explanation: • Replace `REPEAT … UNTIL` with `WHILE … DO … ENDWHILE` • Change condition to `WHILE Number <> 9999 DO` • Add / Move `INPUT Number` to before loop // Move / Add extra `INPUT Number` at end of loop • Remove (`Count ← 0` and) `Count ← Count + 1` Or Any **four** from: • Include an `IF` statement after `INPUT Number` / before updating the arrays • `IF Number <> 9999 THEN …` or similar • Move output statements to be executed when `Number = 9999` • Change `UNTIL Count >= 50` to `UNTIL Number = 9999` • Remove (`Count ← 0` and) `Count ← Count + 1`	**4**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(a)</td><td>
<p><b>One</b> mark for correct input (<b>all sets</b>)
<b>One</b> mark for correct calculations (<b>all sets</b>)
<b>One</b> mark for <b>each</b> correct output</p>
<p><b>Set 1:</b> 88, 74, 60</p>
<table>
<tr><th>Mark1</th><th>Mark2</th><th>Mark3</th><th>Total</th><th>Average</th><th>OUTPUT</th></tr>
<tr><td>88</td><td>74</td><td>60</td><td>222</td><td>74</td><td>Pass</td></tr>
</table>
<p><b>Set 2:</b> 20, 33, 67</p>
<table>
<tr><th>Mark1</th><th>Mark2</th><th>Mark3</th><th>Total</th><th>Average</th><th>OUTPUT</th></tr>
<tr><td>20</td><td>33</td><td>67</td><td>120</td><td>40</td><td>Pass</td></tr>
</table>
<p><b>Set 3:</b> 79, 91, 70</p>
<table>
<tr><th>Mark1</th><th>Mark2</th><th>Mark3</th><th>Total</th><th>Average</th><th>OUTPUT</th></tr>
<tr><td>79</td><td>91</td><td>70</td><td>240</td><td>80</td><td>Distinction</td></tr>
</table>
</td><td><b>5</b></td></tr>
<tr><td>3(b)</td><td>
<p>Any <b>three</b> from:</p>
<ul>
<li>Add extra decision box …</li>
<li>… in an appropriate position between the average calculation and the output</li>
<li>Check for average greater than or equal to 60 and less than 80</li>
<li>Output Merit if average greater than or equal to 60 (and less than 80) …</li>
<li>… otherwise continue</li>
</ul>
</td><td><b>3</b></td></tr>
</table>

Question	Answer	Marks
4	**One** mark for explaining why a validation check is needed when data is input To check that data is sensible / reasonable / meets required criteria **One** mark for explaining why a verification check is needed when data is input To check that data is not changed on entry **One** mark for an example of a validation check Range check // Length check // Type check **One** mark for an example of a verification check Double entry // Visual check	**4**

Question	Answer	Marks
5(a)	One mark for data type and reason SIZE text, expressed as a single word SHAPE text, short phrase required WOOD text, expressed as a single word PRICE currency, needs to be expressed as dollars / may be used in calculations SOLD Boolean, only two choices	**5**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(b)(i)</td><td><b>One mark</b> for
• correct field and table names must include SIZE, WOOD and PRICE, allow SHAPE or SOLD
• correct show for PRICE only
• correct criteria to select small and walnut
<table>
<tr><td>Field:</td><td>SIZE</td><td>WOOD</td><td>PRICE</td><td></td></tr>
<tr><td>Table:</td><td>BOX</td><td>BOX</td><td>BOX</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>=’small’</td><td>=’walnut’</td><td></td><td></td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td><td>3</td></tr>
<tr><td>5(b)(ii)</td><td>
<table>
<tr><td>Field:</td><td>WOOD</td><td></td></tr>
<tr><td>Table:</td><td>BOX</td><td></td></tr>
<tr><td>Sort:</td><td></td><td></td></tr>
<tr><td>Show:</td><td>☑</td><td>(1)</td></tr>
<tr><td>Criteria:</td><td>=’walnut’ OR ‘beech’</td><td>(1)</td></tr>
<tr><td>or:</td><td></td><td></td></tr>
</table>
One mark for each correct alteration max 2</td><td>2</td></tr>
</table>