



Cambridge IGCSE™

COMPUTER SCIENCE**0478/21**

Paper 2

May/June 2020

MARK SCHEME

Maximum Mark: 50

Published

Students did not sit exam papers in the June 2020 series due to the Covid-19 global pandemic.

This mark scheme is published to support teachers and students and should be read together with the question paper. It shows the requirements of the exam. The answer column of the mark scheme shows the proposed basis on which Examiners would award marks for this exam. Where appropriate, this column also provides the most likely acceptable alternative responses expected from students. Examiners usually review the mark scheme after they have seen student responses and update the mark scheme if appropriate. In the June series, Examiners were unable to consider the acceptability of alternative responses, as there were no student responses to consider.

Mark schemes should usually be read together with the Principal Examiner Report for Teachers. However, because students did not sit exam papers, there is no Principal Examiner Report for Teachers for the June 2020 series.

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

Cambridge International is publishing the mark schemes for the June 2020 series for most Cambridge IGCSE™ and Cambridge International A & AS Level components, and some Cambridge O Level components.

This document consists of **10** 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
1(a)(i)	Any meaningful array related to Task 1 – one mark (max two) e.g. ItemCode Description Correct purpose for each array related to Task 1 – one mark (max two) e.g. ...to store the item codes ...to store the descriptions of the items for sale	4
1(a)(ii)	Any meaningful variable related to Task 1 – one mark (max two) e.g. PurchaseItem TotalPrice Correct purpose for each variable related to Task 1 – one mark (max two) e.g. ... to allow input of an item/code for purchase ... to store/calculate the total price of the transaction	4
1(b)	Any one correct statement e.g. - The Code data is made up of letters/alphabetic characters/not numbers - Real data must have numerical value/would not be used in calculations One mark for: - String	2

Question	Answer	Marks
1(c)	Any five from • Input with message to allow choice of mobile device • Attempt to validate input to check for valid item code • Accurate validation of input to check for valid item code • Determination of whether device is phone or tablet • Restriction to only allow input for SIM card required if mobile device is a phone • Input with message to find out if a SIM card is required Example answer <pre> OUTPUT "Which type of phone or tablet would you like? Input the Item Code" DeviceFlag ← False WHILE DeviceFlag = False INPUT DeviceCode Count ← 0 WHILE Count < 10 DO IF DeviceCode = ItemCode[Count] THEN IF Count < 6 THEN DeviceType ← "Phone" ELSE DeviceType ← "Tablet" ENDIF DeviceFlag ← True Count ← 10 ENDIF Count ← Count + 1 ENDWHILE IF DeviceFlag = False THEN </pre>	5

Question	Answer	Marks
1(c)	<pre> OUTPUT "Your code doesn't exist, please try again" ENDIF ENDWHILE IF DeviceType ← "Phone" THEN OUTPUT "Would you like a SIM Card? (Answer Y or N)" INPUT SimRequired ENDIF </pre>	
1(d)	Any five from • Explanation of finding if more than one device is purchased • Explanation of application of device discount • Explanation of calculating the discount(s) and finding the new price • Explanation of outputting new total • Explanation of outputting amount saved • Explanation of messages used	5

Question	Answer	Marks															
2	<table border="1"> <thead> <tr> <th>Statement</th><th>True (✓)</th><th>False (✓)</th></tr> </thead> <tbody> <tr> <td>A structure diagram is a piece of code that is available throughout the structure of a program.</td><td></td><td>✓</td></tr> <tr> <td>A structure diagram shows the hierarchy of a system.</td><td>✓</td><td></td></tr> <tr> <td>A structure diagram is another name for an array.</td><td></td><td>✓</td></tr> <tr> <td>A structure diagram shows the relationship between different components of a system.</td><td>✓</td><td></td></tr> </tbody> </table> Two marks for four correct rows. One mark for three correct rows.	Statement	True (✓)	False (✓)	A structure diagram is a piece of code that is available throughout the structure of a program.		✓	A structure diagram shows the hierarchy of a system.	✓		A structure diagram is another name for an array.		✓	A structure diagram shows the relationship between different components of a system.	✓		2
Statement	True (✓)	False (✓)															
A structure diagram is a piece of code that is available throughout the structure of a program.		✓															
A structure diagram shows the hierarchy of a system.	✓																
A structure diagram is another name for an array.		✓															
A structure diagram shows the relationship between different components of a system.	✓																

Question	Answer	Marks
3(a)	One mark for each correct validation check (max two) • Range • Length • Type • Check Digit One mark for each correct related purpose (max two) e.g. • To make sure the data entered falls within a specific set of values • To make sure the data entered is no longer than specified • To make sure the data entered follows rules related to whether it is numbers or letters • To make sure an identification code entered is genuine or possible	4
3(b)	One mark for correct verification check (max one) • Double (data) entry • Visual check	1
3(c)	Any two correct statements (max two) e.g. • Validation checks if the data entered is possible/it cannot check if data has been entered correctly. • Verification checks if the data entered matches the data submitted for entry/ it does not check if data matches set criteria.	2

Question	Answer	Marks
4(a)	Any two correct statements (max two) e.g. • The value of the variable Count begins as 0 ... • ... and is incremented by 1 before it is tested by the loop condition • Count will never be 0 at the end of the loop	2

Question	Answer	Marks
4(b)	<pre> Count ← 0 REPEAT INPUT Number IF Number >= 100 THEN Values[Count] ← Number ENDIF Count ← Count + 1 UNTIL Count = 50 </pre> One mark – separate INPUT statement One mark – IF statement attempted One mark – IF statement completely correct One mark – termination of loop updated	4
4(c)	Any two correct statements (max two) e.g. • Alter the IF statement/add a second IF statement/comparison that's already there ... • ... so that additional criteria set an upper limit of ≤ 200	2

0478/21

Question	Answer	Marks																																												
5(a)	<table><thead><tr><th>Value</th><th>Calc1</th><th>Calc2</th><th>OUTPUT</th></tr></thead><tbody><tr><td>50</td><td>25</td><td>16</td><td></td></tr><tr><td>33</td><td>16</td><td>11</td><td></td></tr><tr><td>18</td><td>9</td><td>6</td><td>18</td></tr><tr><td>15</td><td>7</td><td>5</td><td></td></tr><tr><td>30</td><td>15</td><td>10</td><td>30</td></tr><tr><td>-1</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table> One mark for each correct column (max four)	Value	Calc1	Calc2	OUTPUT	50	25	16		33	16	11		18	9	6	18	15	7	5		30	15	10	30	-1																				4
Value	Calc1	Calc2	OUTPUT																																											
50	25	16																																												
33	16	11																																												
18	9	6	18																																											
15	7	5																																												
30	15	10	30																																											
-1																																														
5(b)	Any two correct statements e.g. - The program outputs a value - That is divisible by 6 // 2 and 3	2																																												

Question	Answer					Marks																																				
6(a)	To uniquely identify a product (in TOOLS)					1																																				
6(b)	<table><tr><td>HS50</td><td>Hose (50 metres)</td><td>60</td></tr><tr><td>GFLG</td><td>Garden Fork</td><td>50</td></tr><tr><td>LMPT</td><td>Lawn Mower (Petrol)</td><td>25</td></tr></table> One mark for correct data One mark for correct format One mark for data in correct order					HS50	Hose (50 metres)	60	GFLG	Garden Fork	50	LMPT	Lawn Mower (Petrol)	25	3																											
HS50	Hose (50 metres)	60																																								
GFLG	Garden Fork	50																																								
LMPT	Lawn Mower (Petrol)	25																																								
6(c)	<table><tr><td>Field:</td><td>Code</td><td>Description</td><td>Quantity_Stock</td><td></td><td></td></tr><tr><td>Table:</td><td>TOOLS</td><td>TOOLS</td><td>TOOLS</td><td></td><td></td></tr><tr><td>Sort:</td><td>Ascending</td><td></td><td></td><td></td><td></td></tr><tr><td>Show:</td><td>☒</td><td>☒</td><td>☒</td><td>☐</td><td>☐</td></tr><tr><td>Criteria:</td><td></td><td></td><td><25</td><td></td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td><td></td></tr></table> One mark for each completely correct column (max three)					Field:	Code	Description	Quantity_Stock			Table:	TOOLS	TOOLS	TOOLS			Sort:	Ascending					Show:	☒	☒	☒	☐	☐	Criteria:			<25			or:						3
Field:	Code	Description	Quantity_Stock																																							
Table:	TOOLS	TOOLS	TOOLS																																							
Sort:	Ascending																																									
Show:	☒	☒	☒	☐	☐																																					
Criteria:			<25																																							
or:																																										