Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/22**

Paper 2 **October/November 2019**

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 2019 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

This syllabus is regulated for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

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.

Section A

Question	Answer	Marks
1(a)(i)	Data Structure(s) maximum two marks • Array • Variable • Constant Description maximum three marks • Name(s) one or more e.g. Colour • Data type(s) one or more e.g. String • Use(s) one or more e.g. To store the colour of the slab Additional data structure description using the same data structure type maximum one mark • Two or more full descriptions of the data structure including name, data type and use	**5**
1(a)(ii)	• Variable name e.g. ConcretePrice • Data type e.g. Real • Use e.g. To store the price of the grade of concrete	**3**
1(b)	**Three** from: • Using the shape, size and depth … • … the area of the shape is found … • … for all except round multiply length by breadth/show example … • … for round multiply Pi by half the diameter/radius squared … • … multiply the area by the depth to give the volume	**3**

Question	Answer	Marks

1(c)	**Five** from:
	MP1 Enter number of slabs to purchase with prompt
	MP2 Check if less than 20 or greater than 100 …
	MP3 … if so, reject the number and re-enter a value
	MP4 Attempt to round the number of slabs to the next 20 …
	MP5 Accurate attempt to round up to the next 20
	MP6 Calculate the number of batches
	MP7 Calculate price to pay (number of batches × price per batch from Task 1)
	MP8 Display price and number of slabs produced …
	MP9 … Suitable output message(s)
	
	Sample answer
	`REPEAT`
	`  PRINT "Enter Number of slabs to purchase"`
	`  INPUT Number`
	`UNTIL Number >= 20 and Number <= 100`
	`PurchaseNo ← Number`
	`IF Number > 20 AND Number <= 40 THEN PurchaseNo ← 40 ENDIF`
	`IF Number > 40 AND Number <= 60 THEN PurchaseNo ← 60 ENDIF`
	`IF Number > 60 AND Number <= 80 THEN PurchaseNo ← 80 ENDIF`
	`IF Number > 80 THEN PurchaseNo ← 100 ENDIF`
	`BatchNo ← PurchaseNo / 20`
	`PriceToPay ← BatchNo * BatchPrice //BatchPrice calculated in Task 1`
	`PRINT "Price for ", PurchaseNo, " Slabs is $", PriceToPay`
		5

1 (d)	**Four** from explanations: • Enter the price of the concrete • Store the price of the concrete • Set up variable(s) for grades • Input the grade • The price that is input is used in the final price calculation instead of 0.05 • The price calculation depends on the grade input e.g. 1.07 × price or price + price × 0.07 for Best If no programming code seen to support at least one explanation maximum of three marks can be awarded.	**4**

Section B

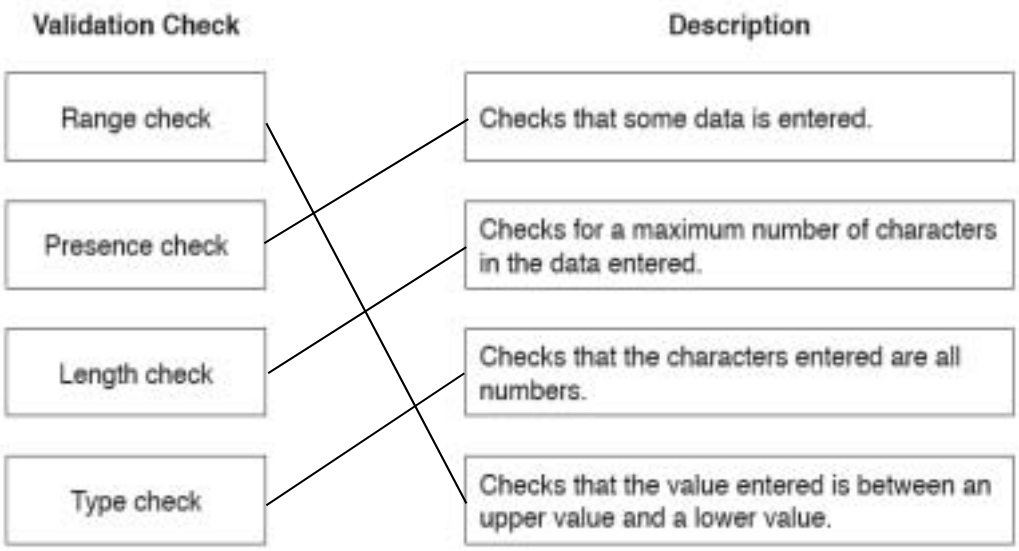
Question	Answer	Marks
2	**One** mark for each error identified and suggested correction: • `IF TotalTry > Number` should be `IF Guess > Number` • `IF Guess > Number` should be `IF Guess < Number` • `TotalTry ← Guess + 1` should be `TotalTry ← TotalTry + 1` • `UNTIL Guess <> Number` should be `UNTIL Guess = Number`	**4**

Question	Answer	Marks
3(a)		**4**

Reject	Count	Length	OUTPUT
0	0		
0	1	24.88	
1	2	25.01	
1	3	24.98	
1	4	25.00	
1	5	25.05	
1	6	24.99	
1	7	24.97	
1	8	25.04	
1	9	25.19	
2	10	25.07	Batch rejected

One mark for each correct column **max. 4**

3(b)(i)	• Remove `Length < 25.1 AND`	**1**
3(b)(ii)	**Two** from: • Check the reject counter after each incrementation/remove reject check after counter = 10 … • … as soon as Reject = 2 / >1 ... • … reject batch and end	**2**

Question	Answer	Marks
4	 One mark for each correct line, up to maximum of three marks.	**3**

Question	Answer	Marks
5	Totalling: • Adding the weight of each basket to the total weight as each weight is entered • `Total = Total + Weight` Counting: • Adding one to/incrementing the number of baskets as each weight is entered • `BasketCount = BasketCount + 1`	**4**

Question	Answer	Marks
6	Constants **Two** from: • The value cannot be changed accidentally … • … during the **execution** of the program • Value only needs to be changed once if circumstances change/during the initialisation process Variables **Two** from: • Stores a value that can change … • … during the **execution** of the program • Can use a variable without knowing its value Arrays **Two** from: • A list of items of the same data type … • … stored under a single name • To reduce the number of variables used • Any item can be found using an index number to show its place in the list	**6**

Question	Answer	Marks
7(a)	• Number is repeated/not unique	**1**
7(b)	• Item number not displayed/Amount column not required • Not Like 'Delivered' will also show cancelled items • Correct Item number column • Correct Order number column and any additional column not shown • Correct status column	**5**

Field:	Item number	Order number	Status	
Table:	SALES	SALES	SALES	
Sort:				
Show:	☑	☑	☐	☐
Criteria:			Like "Not started"	
or:			Like "In progress"	