Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/22**

Paper 2 **March 2018**

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 2018 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

® IGCSE is a registered trademark.

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

© UCLES 2018

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 2018

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.

© UCLES 2018

Question	Answer	Marks
	Section A	
1(a)(i)	1 mark for name and 1 mark for appropriate data structure from **Task 1** e.g. name `studentName` Data structure Array	**2**
1(a)(ii)	1 mark for name, 1 mark for appropriate value and 1 mark for appropriate use. From any task e.g. Constant name `classSize` Value 30 Use Fixed number of students in a class Variable name `counter` Data type integer Use to count the number of student names entered	**6**
1(b)	1 mark for changing the value of a variable/constant/ counter, 1 mark for change to the program e.g. Value: Change the value of the variable/constant `maxGroupSize` in **Task 2** to 25 Program change: Oversubscribed test will be changed to test for over 50 // second group starts at 26	**2**

© UCLES 2018

Question	Answer	Marks
1(c)	Any **five** from Initialise subject counts Loop through students Input and store student name… … and input and store two subject choices for that student Update subject counts // update subject lists of student names Error message if subject not found Output total subject counts Prompts for all inputs and messages with all outputs Sample answer for **Task 1**	**5**

```
physics ← 0; chemistry ← 0; history ← 0; geography ← 0; compSci ← 0;
FOR counter ← 1 TO 60
   INPUT name, subject(1), subject(2)
   studentName(counter) ← name
   studentCounter1(counter) ← subject(1)
   studentCounter2(counter) ← subject(2)
   FOR sCount ← 1 TO 2
      CASE subject(sCount) OF
            'Physics': physics ← physics + 1
            'Chemistry': chemistry ← chemistry + 1
            'History': history ← history + 1
            'Geography': geography ← geography + 1
            'Computer Science': compSci ← compSci + 1
      OTHERWISE: PRINT 'Error subject not found'
      ENDCASE
   NEXT
NEXT
PRINT 'Group            Number of Students'
PRINT 'Physics          ',physics
PRINT 'Chemistry        ',chemistry
PRINT 'History          ',history
PRINT 'Geography        ',geography
PRINT 'Computer Science ',compSci
```

Question	Answer	Marks
1(d)	Any **five** from Explanation for calculation of total number of spare places Check the number of students who have chosen each subject If less than 10 no places available (Otherwise)If less than 20… …Calculate number of spare places (20 – number) If more than 20 and less than 40… …Calculate number of spare places (40 – number) Keep a running total of the total number of spare places	**5**

© UCLES 2018

Question	Answer	Marks
	Section B	

2(a)	1 mark for each error identified + suggested correction
	`NUMBERS` should be `Number`
	`IF Number > 100` should be `IF Number >= 100`
	`INPUT Number` is missing from inside the loop insert `INPUT Number` after the `IF` statement
	The final `PRINT Number` is not needed remove it
		4

2(b)	One mark for both ends of the range and correct inequality symbols and one mark for the **AND.** The test should be `IF Number >= 100 AND Number <= 200`	**2**

Question	Answer	Marks
3		**4**

Weight	totalWeight	totalNumber	OUTPUT
	0	0	
50	50	1	
70	120	2	
65	185	3	
100	285	4	
95	380	5	
50	430	6	
55	485	7	
85	570	8	
70	640	9	Lift overload, step out

One mark for each correct column.

© UCLES 2018

Question	Answer	Marks
4	1 mark for value and 1 mark for appropriate reason e.g. Value 1 `2` (1) boundary should be accepted as weight OK (1) Value 2 `two` (1) erroneous/abnormal should be rejected (1)	**4**

Question	Answer	Marks

5	2 marks for appropriate explanation, 1 mark example programming statements showing sequence, 1 mark example programming statement(s) showing selection e.g.
	
	Sequence is the concept of one statement being executed after another(1) whereas selection decides which statement(s) are to be executed depending upon the result of a question (1)
	sequence example (1)
	`PRINT X`
	`PRINT Y`
	Selection example (1)
	`IF X > Y THEN PRINT X ELSE PRINT Y`
		4

Question	Answer	Marks
6(a)	1 mark for appropriate field name and appropriate data type, then 1 mark for appropriate explanation. e.g. Metal, type text(1) a single character/word that can be input accurately/quickly(1) Item, type text (1) a single character/word that can be input accurately/quickly(1) Number in Stock, type number (1) can be used for calculations (1) Price, type currency (1) properly formatted and can be used for calculations (1)	**8**
6(b)	All fields could contain duplicate values	**1**

© UCLES 2018

Question	Answer	Marks
6(c)		**3**

Field:	Metal	Item	Number in Stock	Price
Table:	JEWEL	JEWEL	JEWEL	JEWEL
Sort:				
Show:	☐	☐	☑	☑
Criteria:	="silver"	="bracelet"		
or:				

One mark for columns 1 + 2, one mark for columns 3 + 4
One mark for accuracy of syntax and spelling

© UCLES 2018