Cambridge
O Level

**Cambridge International Examinations**
Cambridge Ordinary Level

**COMPUTER SCIENCE** **2210/21**

Paper 2 **May/June 2016**

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 will not enter into discussions about these mark schemes.

Cambridge is publishing the mark schemes for the May/June 2016 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

## Section A

**1 (a) (i) Many correct answers, names must be meaningful. This is an example only.**

`Length`, real/integer, length of parcel
`Breadth`, real/integer, breadth of parcel
`Height`, real/integer, Height of parcel [3]

**(ii) Several correct answers, they must be meaningful. These are examples only.**

```
Dimension, 80
TotalDimension 200
MaxWeight 10.00
```
[2]

**(b)** Any **5** from:

– input length, breadth, height and weight
– check each dimension, not more than 80
– check total of dimensions, not more than 200
– check weight at least 1
– check weight not more than 10
– output parcel accepted (must be in appropriate position)
– output parcel rejected (must be in appropriate position)
– output all reasons for rejecting parcel (reason must follow test) [5]

Max 5 marks

Sample Answer.

```
INPUT Length, Breadth, Height, Weight
IF Length <= 80 AND Breadth <= 80 AND Height <= 80 AND Weight >= 1
AND Weight <=10 AND Length + Breadth + Height <= 200 THEN
  PRINT 'Parcel accepted'
  ELSE
    PRINT 'Parcel rejected'
    IF Length > 80 OR Breadth > 80 OR Height > 80 THEN
      PRINT 'At least one dimension too large'
    ENDIF
    IF Weight < 1 THEN
      PRINT 'Parcel too light'
    ENDIF
    IF Weight > 10 THEN
     PRINT 'Parcel too heavy'
    ENDIF
ENDIF
```

**(c)** **1** mark for the data set and **1** mark for the matching reason all, data sets and reasons must be different. There are many possible correct answers these are examples only.

Data set 30, 29, 28, 4
Reason – normal data; parcel should be accepted

Data set 80, 60, 60, 10
Reason – boundary data; parcel should be accepted

Data set – 85, 60, 60, 11
Reason – abnormal data; parcel should be rejected [6]

**(d)** Maximum **4** marks in total, maximum **2** marks if only programming statements used**.**

Explanation (may include reference to programming statements)
– loop for number of parcels
– parcels 5 kg or less use standard price
– over 5 kg use weight to calculate price
– Correct calculation of price
– keep running total of consignment price [4]

## Section B

**2** **(i)** **1** mark for each change

Change variable name in every instance as needs to be meaningful e.g. `Large`
Set this variable to a low value
line 5: change comparison from < to > [3]

**(ii)** **3** marks maximum, 1 mark for each change correctly included.

```
1  Large = 0
2  Counter = 0
3  REPEAT
4    INPUT Num
5    IF Num > Large THEN Large = Num
6    Counter = Counter + 1
7  UNTIL Counter = 10
8  PRINT Large
```

[3]

**3** **(i)** Name type – string
Gender type – char/string
Status type – char/string
Fee type – real
Team member type – Boolean [5]

**(ii)** Data Structure – several Arrays ……
…….Reason – to simplify programming/ make programs shorter/index can be used to identify the same member across the arrays etc.

[2]

**4**

| Riders | Reject | Height | Output |
|---|---|---|---|
| 0 | 0 | | |
| 1 | | 1.4 | |
| 2 | | 1.3 | |
| | 1 | 1.1 | |
| 3 | | 1.3 | |
| | 2 | 1.0 | |
| 4 | | 1.5 | |
| 5 | | 1.2 | |
| 6 | | 1.3 | |
| 7 | | 1.4 | |
| 8 | | 1.3 | |
| | | | Ready to go 2 |
| (1 mark) | (1 mark) | (1 mark) | (1 mark) |

[4]

**5** – `FOR (… TO … NEXT)…`
– … a set number of iterations
– `WHILE (… DO … ENDWHILE) …`
– … used where the loop may never be executed/whilst a specified condition exists

[4]

**6** **(a)** – all (fields) have (1 mark) duplicate entries (1 mark)

– none (of the fields) (1 mark) have unique entries(1 mark)

[2]

**(b)** – e.g. StaffNumber ….

– ….. Uniquely identifies each member of staff//no duplicates//different for each member of staff

[2]

**(c)**

| | | | | |
|---|---|---|---|---|
| Field: | Department | Name | | |
| Table: | STAFFPHONE | STAFFPHONE | | |
| Sort: | Ascending | Ascending | | |
| Show: | ☑ | ☑ | ☐ | ☐ |
| Criteria: | | | | |
| or: | | | | |

(2 marks) (2 marks) (1 mark for correct order and number of fields shown) [5]