MODULE M PROGRAMMING · M28

Writing maintainable code: identifiers, comments and sub programs

Original practice questions · 20 marks · ~25 minutes · Cambridge 0478 syllabus 8.1.8, 8.1.6, 8.1.1

Name _____ Date _____

Instructions. Answer all questions in the spaces provided. The marks for each part are shown in brackets [ ]. The total for this worksheet is **20 marks**. All algorithms on this sheet are written in pseudocode. Write your own algorithms and statements in pseudocode.

1

Total: 2 marks

(a) Tick (✓) **two** boxes to show which techniques are used to create a maintainable program. **[2]**

	Technique	Tick (✓)
A	Using comments to explain what parts of the program do	
B	Using a range check on each input	
C	Using meaningful identifiers	
D	Using double entry to verify data	
E	Using a trace table to dry-run the algorithm	

2 **Total: 8 marks**

A library charges $0.25 for each day a book is returned late, up to a maximum fee of $5.00. This algorithm works out the fee.

```
01 CONSTANT R ← 0.25
02 CONSTANT M ← 5.0
03 DECLARE D : INTEGER
04 DECLARE F : REAL
05 OUTPUT "Charge per day late is $", R
06 INPUT D
07 F ← D * R
08 IF F > M
09   THEN
10     F ← M
11 ENDIF
12 OUTPUT "Fee to pay is $", F
```

(a) Rewrite lines 06 to 11 using meaningful identifiers in place of `R`, `M`, `D` and `F`. **[4]**

(b) Explain why using meaningful identifiers makes this algorithm easier to maintain. **[2]**

(c) The library changes the charge for each day from $0.25 to $0.30. **[2]**
Explain why storing the charge as a constant with a meaningful identifier makes this change easier to make.

3 **Total: 7 marks**

A garden centre delivers parcels. The delivery charge is $1.50 for each kilogram, with a minimum charge of $4.00. This algorithm works out the total delivery charge for two parcels. The variables are all of data type REAL; their declarations are not shown.

```
01 INPUT Weight1
02 Charge1 ← Weight1 * 1.5
03 IF Charge1 < 4.0
04   THEN
05     Charge1 ← 4.0
06 ENDIF
07 INPUT Weight2
08 Charge2 ← Weight2 * 1.5
09 IF Charge2 < 4.0
10   THEN
11     Charge2 ← 4.0
12 ENDIF
13 OUTPUT "Total delivery charge is $", Charge1 + Charge2
```

A programmer decides to replace the repeated code with a function called `DeliveryCharge`.

(a) Write pseudocode to define the function `DeliveryCharge`. The function: **[4]**

- takes the weight of one parcel, in kilograms, as a parameter
- calculates the charge at $1.50 for each kilogram
- uses a charge of $4.00 if the calculated charge is less than $4.00
- returns the charge.

(b) Write a pseudocode statement to call the function using `Weight1` as the parameter and store the return value in `Charge1`. This statement will replace lines 02 to 06. **[1]**

(c) Explain **one** way that using the function makes this program easier to maintain. **[2]**

4 **Total: 3 marks**

A weather station stores 50 temperature readings, in °C, in the 1D array `Temperature`. This algorithm counts how many of the readings are below zero.

```
01 Count ← 0
02 FOR Index ← 1 TO 50
03     // add 1 to Count
04     IF Temperature[Index] < 0
05       THEN
06         Count ← Count + 1
07     ENDIF
08 NEXT Index
09 OUTPUT "Readings below zero: ", Count
```

(a) Explain why the comment on line 03 is **not** a useful comment. **[2]**

(b) Write a more useful comment, in pseudocode, that could replace line 03. **[1]**

Total for this worksheet: 20 marks