2.4 Programming Paradigms and OOP: 10-minute Pack

Total Marks

OCR H446 · Calculators not allowed · about 10 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 A small program for a Bristol newsagent totals up the cost of a customer's magazines. It is written using OCR's Exam Reference Language conventions (assignment <-, iteration FOR ... NEXT, selection IF ... ENDIF).

```
PROCEDURE CalculateTotal(price, quantity)
  total <- price * quantity
  RETURN total
ENDPROCEDURE

PROCEDURE PrintReceipt(item, price, quantity)
  total <- CalculateTotal(price, quantity)
  OUTPUT item, " x ", quantity, " = GBP ", total
ENDPROCEDURE

CALL PrintReceipt("Weekly Puzzle Magazine", 2.50, 4)
```

(a) Identify the programming paradigm used in the code above. **(1)**

(b) Give two reasons for your answer to part (a), referring directly to the code. **(2)**

(c) State one change that would need to be made to this code for it to become object-oriented. **(1)**

(Total for Question 1 is 4 marks)

© Revision Library | revisionlibrary.co.uk

2 A secondary school in Norwich wants a pseudocode class to represent a book held in its school library.

(a) Write pseudocode for a class LibraryBook with four private attributes: title (STRING), author (STRING), isbn (STRING) and isOnLoan (BOOLEAN); and a constructor PROCEDURE NEW(t, a, i) that sets title, author and isbn from its parameters and sets isOnLoan to FALSE. **(3)**

(b) Write a public procedure BorrowBook() for LibraryBook that sets isOnLoan to TRUE and outputs a confirmation message if the book is not already on loan; otherwise it should output a message stating the book is already on loan and leave isOnLoan unchanged. **(3)**

(Total for Question 2 is 6 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 2.4 Programming Paradigms and OOP: 10-minute Pack

Question 1

(a) procedural (programming) cao **B1**

(a) Answer: Procedural programming.

(b) the code is organised as procedures (CalculateTotal, PrintReceipt) that are called in sequence/one from another (oe) **B1**

(b) there are no classes or objects defined; data (price, quantity, total) is passed directly as parameters rather than being bundled with methods (oe) **B1**

(b) Answer: The code uses two procedures that call each other in sequence, and there are no classes or objects; data is simply passed between procedures as parameters.

(c) define a class (e.g. Receipt or Item) that bundles the item's data (item, price, quantity) together with methods such as CalculateTotal and PrintReceipt (oe) **B1**

(c) Answer: Create a class, for example Receipt, whose attributes hold item, price and quantity, and whose methods include CalculateTotal and PrintReceipt.

Question 2

(a) declares all four attributes as PRIVATE with appropriate types (title, author, isbn as STRING; isOnLoan as BOOLEAN) **M1**

(a) constructor correctly assigns the three parameters to title, author and isbn **M1**

(a) constructor correctly initialises isOnLoan to FALSE; overall class syntax logically complete (ignore minor syntax errors) **A1**

(a) Answer:

```
CLASS LibraryBook
  PRIVATE title : STRING
  PRIVATE author : STRING
  PRIVATE isbn : STRING
  PRIVATE isOnLoan : BOOLEAN

  PUBLIC PROCEDURE NEW(t, a, i)
    title <- t
    author <- a
    isbn <- i
    isOnLoan <- FALSE
  ENDPROCEDURE
ENDCLASS
```

(b) uses an IF statement to test isOnLoan = FALSE (or NOT isOnLoan) **M1**

(b) TRUE branch sets isOnLoan <- TRUE and outputs a confirmation message **M1**

(b) ELSE branch outputs a message that the book is already on loan and does not change isOnLoan; overall logic complete (ignore minor syntax errors) **A1**

(b) Answer:PUBLIC PROCEDURE BorrowBook()

```
  IF isOnLoan = FALSE THEN
    isOnLoan <- TRUE
    OUTPUT "Book borrowed successfully"
  ELSE
    OUTPUT "This book is already on loan"
  ENDIF
ENDPROCEDURE
```


© Revision Library | revisionlibrary.co.uk