2.2 Problem Solving and Programming: 10-minute Pack

Total Marks

OCR H446 · Calculators not allowed · about 10 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 A game awards a bonus based on a player's grade, using the following CASE statement:

```
CASE OF grade
    "A" : bonus <- 50
    "B" : bonus <- 30
    "C" : bonus <- 10
    OTHERWISE : bonus <- 0
```

ENDCASE

(a) State the value assigned to bonus when grade = "B". **(1)**

(b) Rewrite the CASE statement above as an equivalent structure using nested IF...THEN...ELSE statements. **(4)**

(c) Explain one advantage of using a CASE statement rather than nested IF statements in this situation. **(2)**

(d) A trainee programmer writes this version of the CASE statement: **(2)**

```
CASE OF grade
    "A" : bonus = 50
    "B" : bonus <- 30
    OTHERWISE : bonus <- 0
```

ENDCASE

Identify the error in this code and state how to fix it.

(Total for Question 1 is 9 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 2.2 Problem Solving and Programming: 10-minute Pack

Question 1

(a) bonus = 30, cao **B1**

(a) Answer: 30

(b) IF grade = "A" THEN bonus <- 50 **M1**

(b) nested ELSE IF grade = "B" THEN bonus <- 30 **M1**

(b) nested ELSE IF grade = "C" THEN bonus <- 10 **M1**

(b) final ELSE bonus <- 0, with all IF structures correctly closed with ENDIF; any correct logic, ignore minor syntax **A1**

(b) Answer:

```
IF grade = "A" THEN
    bonus <- 50
ELSE
    IF grade = "B" THEN
        bonus <- 30
    ELSE
        IF grade = "C" THEN
            bonus <- 10
        ELSE
            bonus <- 0
        ENDIF
    ENDIF
ENDIF
```

(c) identifies benefit, e.g. CASE is more readable/easier to follow when testing many discrete values of the same variable, oe **B1**

(c) expands the point, e.g. avoids deeply nested IF statements, which are harder to trace, maintain and debug, oe **B1**

(c) Answer: CASE is clearer than deeply nested IF statements when checking several discrete values of one variable, reducing the chance of logic errors.

(d) identifies that the "A" branch uses "=" instead of the assignment operator **B1**

(d) states the correction: bonus <- 50 **B1**

(d) Answer: The line "A" : bonus = 50 uses "=" (comparison) instead of "<-" (assignment); it should read "A" : bonus <- 50

© Revision Library | revisionlibrary.co.uk