MODULE M PROGRAMMING · M24

Parameters and arguments: mark scheme

20 marks · Cambridge 0478 syllabus 8.1.6(a) (understand what is meant by procedures, functions and parameters), 8.1.6(b) (define and use procedures and functions, with or without parameters)

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Code answers must be in pseudocode; marks are for the features listed, not for one exact answer. Ignore minor slips in case and spacing. Accept `<-` or `=` for assignment. Candidates do not need to use the word "argument": "the value passed (to the parameter)" is equally creditworthy. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 8.1.6(a)	Second row: a variable named in the header of a procedure or function that receives a value when it is called	**1**	More than one tick: 0. Row 1 describes a return value, row 3 a global variable, row 4 a library routine.
2(a) 0478 8.1.6(a)	`NightlyRate` and `Nights`	**1**	Both needed. Ignore data types if given. Do not accept `80.0` and `3` (values passed in the call), `Amount` or `ToPay`.
2(b) 0478 8.1.6(b)	3	**1**	The second value in the call is given to the second parameter.
2(c) 0478 8.1.6(b)	REAL	**1**	Accept real / float. Do not accept INTEGER or "number".
2(d) 0478 8.1.6(b)	`Deposit due: 60`	**1**	Accept 60.0 and accept the message with or without quotation marks. 60 alone (no message) = 0. Working: 80.0 × 3 / 4 = 60.0.
3(a) 0478 8.1.6(b)	One mark for each correct row: • Line 11: `Charge: 4.5` • Line 12: `Charge: 8` // `Charge: 8.0` • Line 13: `Charge: 1` // `Charge: 1.0`	**3**	Ignore quotation marks and minor spacing. Working: 3 × 1.5 = 4.5 (not a member, no reduction); 5 × 2.0 = 10.0 − 2.0 = 8.0; 2 × 1.5 = 3.0 − 2.0 = 1.0. Value with no "Charge:" text: penalise once only.

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) 0478 8.1.6(b)	Any two from: • values in a call are given to the parameters in order / by position • so `Hours` would be given TRUE // `Member` would be given 2.5 // the values go to the wrong parameters • the data types of the values do not match the data types in the procedure definition // a BOOLEAN is given to an INTEGER parameter • the values should be in the order hours, rate, member // `CALL ParkingCharge(4, 2.5, TRUE)`	**2**	Max 2. "The values are in the wrong order" alone = 1. Do not credit "CALL should not be used" (it is a procedure, so CALL is correct) or "there are too many/too few values" (there are three).
4(a) 0478 8.1.6(b)	• `PROCEDURE ShowBadge( … )` with two parameters in brackets • Name parameter typed STRING and desk parameter typed INTEGER Example: `PROCEDURE ShowBadge(VisitorName : STRING, DeskNumber : INTEGER)`	**2**	Any sensible parameter names; accept the parameters in either order. `FUNCTION` in place of PROCEDURE, or a RETURNS clause: max 1. Data type mark needs both types correct. Do not accept TEXT (not a pseudocode data type) or REAL for the desk number. Ignore a missing space either side of the colon.
4(b) 0478 8.1.6(b)	• `CALL ShowBadge( … )` • Values `"Dana Reyes"` and `14` in the same order as the parameters in 4(a) Example: `CALL ShowBadge("Dana Reyes", 14)`	**2**	Follow through the parameter order from 4(a). The name must be in quotation marks. Missing CALL: max 1. Do not accept `"14"` (a string, not an INTEGER) for the second mark. A procedure header written instead of a call: 0.

QUESTION	ANSWER	MARKS	GUIDANCE

5(a)
0478 8.1.6(b)
	One mark per bullet point. Max 4:
	• `FUNCTION InRange(` with three parameters, each typed INTEGER
	• `RETURNS BOOLEAN` in the header
	• Correct inclusive check using the parameters: reading >= lowest `AND` reading <= highest
	• RETURN TRUE and RETURN FALSE in the correct cases // RETURN of the condition, with `ENDFUNCTION`
	Example:
	`FUNCTION InRange(Reading : INTEGER, Lowest : INTEGER, Highest : INTEGER)`
	`RETURNS BOOLEAN`
	`  IF Reading >= Lowest AND Reading <= Highest`
	`  THEN`
	`  RETURN TRUE`
	`  ELSE`
	`  RETURN FALSE`
	`  ENDIF`
	`ENDFUNCTION`
		4	Any sensible parameter names. Accept `RETURN (Reading >= Lowest) AND (Reading <= Highest)`. Accept a correct equivalent such as `NOT (Reading < Lowest OR Reading > Highest)`. > or < (boundaries excluded) loses the check mark. Fixed values (18, 24) in place of the parameters lose the check mark. INPUT statements that overwrite the parameters lose the check mark. OUTPUT in place of RETURN loses the last mark.

5(b) 0478 8.1.6(b)	• `Comfortable ← InRange( … )`: the function is used in an assignment, without CALL • `Temperature, 18, 24` in the same order as the parameters Example: `Comfortable ← InRange(Temperature, 18, 24)`	**2**	Use of CALL with the function loses the first mark. Follow through the parameter order from 5(a). `InRange(18, 24, Temperature)` with the header from the example loses the second mark.

Total: 20 marks