MODULE M PROGRAMMING · M25

Local and global variables: mark scheme

22 marks · OCR J277 spec 2.2.3 (the use of local variables/constants and global variables/constants within functions and procedures; the use of procedures)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. **...** at the end of one bullet and the start of the next means the second mark depends on the first. Code answers are marked for logic, not exact syntax: accept OCR Exam Reference Language or any high-level language, and allow any logically valid method. Minor syntax slips are not penalised; a repeated slip is penalised once then follow through. Credit any other valid answer.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.2.3	Used in the main program and every sub program: Global variable Declared inside a procedure or function: Local variable Only exists while its sub program is running: Local variable	**3**	1 mark per correct row. Two ticks in a row: 0 for that row.
2(a) J277 2.2.3	`totalTakings`	**1**	Correct answer only. Ignore case. Do not accept `global` on its own (the keyword, not the variable).
2(b) J277 2.2.3	`RATE`	**1**	Correct answer only. Ignore case. Accept "RATE = 2" / line 04. Do not accept `fee` or `hours` (local variables, not a constant).

2(c) J277 2.2.3	`Fee: 6`
	`Fee: 10`
	`Total: 16`
		3	1 mark per correct line, in this order. Ignore quotation marks and small spacing differences. A line given as the number only (e.g. 6, 10, 16) = max 2 in total. Working: 3 × 2 = 6; 5 × 2 = 10; 0 + 6 + 10 = 16.

2(d) J277 2.2.3	1 mark each to max 2: `fee` is a local variable / is declared inside the procedure `takePayment` ... ... so it can only be used / accessed inside that procedure // cannot be used in the main program ... `fee` only exists while the procedure is running / no longer exists after the procedure ends (line 08)	**2**	Second and third points need `fee` to be identified as local (or declared in the procedure). "fee has not been declared" alone = NE. "It is a syntax error" alone = 0.

QUESTION	ANSWER	MARKS	GUIDANCE
2(e) J277 2.2.3	1 mark each to max 2: It is changed inside the procedure **and** used in the main program (line 12) / it is needed in more than one part of the program It must keep its value between calls to the procedure / keeps adding up across both payments A local variable would be lost / reset when the procedure ends (so the total could not be kept or printed)	**2**	Answers must refer to this program for the first two points. "So it can be used anywhere" with no link to this program = max 1. Do not accept "because it is a total" alone.
3(a) J277 2.2.3	1 mark for each benefit and 1 for its explanation, max 2 benefits (max 4): A local variable can only be changed inside its own sub program ... ... so other programmers' code cannot accidentally change its value / fewer unexpected side effects The same variable name can be used in different sub programs ... ... without the variables clashing / programmers do not need to check which names others have used Each sub program is self-contained / does not depend on variables outside it ... ... so it can be tested / debugged on its own // can be reused in another program more easily Local variables only exist while the sub program runs ... ... so the memory they use is freed when the sub program ends	**4**	Mark in pairs: an explanation only scores with its benefit. The same benefit given twice scores once. "Easier to debug" with no reason = NE for the explanation mark unless linked to a benefit. Do not accept "faster" or "more secure" alone. Credit any other valid benefit with an explanation.

4(a)
J277 2.2.3
	1 mark each to max 5:
	Procedure defined with the name `addStamps`, a parameter in brackets and an `endprocedure` (or equivalent end of block)
	Stamps worked out as amount `DIV 5` / `int(amount / 5)` / `amount // 5` (or any method that gives the whole number of full £5s) ...
	... and stored in a variable declared inside the procedure (local)
	`totalStamps = totalStamps + stamps` (global updated by adding, not replaced)
	Outputs the number of stamps earned (the local variable)
	Example:
	`procedure addStamps(spend)`
	`    stamps = spend DIV 5`
	`    totalStamps = totalStamps + stamps`
	`    print(stamps)`
	`endprocedure`
		5	Accept a function in place of a procedure. Python: accept `def addStamps(spend):` and accept with or without a `global totalStamps` line inside. Do not award the "stored ... local" mark if the stamps are stored in `totalStamps` or in a variable made global. Setting `totalStamps = 0` inside the procedure loses the update mark. `spend / 5` with no rounding down does not get the calculation mark. A parameter immediately overwritten by an input loses the first mark. Do not accept flowcharts, structured English or bullet points.

QUESTION	ANSWER	MARKS	GUIDANCE
4(b) J277 2.2.3	6	**1**	Correct answer only. Working: 23 DIV 5 = 4; 12 DIV 5 = 2; 0 + 4 + 2 = 6.

Total: 22 marks