MODULE M PROGRAMMING · M25

# Local and global variables: mark scheme

**22 marks · Cambridge 0478 syllabus 8.1.6(c) (understand and use local and global variables), 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. **...** at the start of a bullet means the point builds on the one before. Algorithm answers must be in pseudocode; marks are for the features listed, not for one exact algorithm. Ignore minor slips in case and spacing. Accept <- or = for assignment. Credit any other valid answer that shows the same understanding.

| QUESTION | ANSWER | MARKS | GUIDANCE |
|---|---|---|---|
| **1(a)** 0478 8.1.6(c) | One mark for each correct row:<br>• declared inside a procedure or function, only used there: Local variable<br>• declared in the main program, used anywhere: Global variable<br>• memory freed when the procedure ends: Local variable | **3** | Two ticks in a row: no mark for that row. |
| **2(a)** 0478 8.1.6(c) | `TotalMinutes` | **1** | Ignore case. |
| **2(b)** 0478 8.1.6(c) | (Line) 02 | **1** | Accept 2. Also accept 01 / 1 (Minutes is a parameter, which is local to EndTrip). Do not accept line 09 (declares the global variable). |
| **2(c)** 0478 8.1.6(b), (c) | One mark for each correct output, in order:<br>• `Credits used: 30`<br>• `Credits used: 16`<br>• `Total minutes: 23` | **3** | Ignore quotation marks, commas and small spacing differences. Working: 15 × 2 = 30; 8 × 2 = 16; 0 + 15 + 8 = 23. |
| **2(d)** 0478 8.1.6(c) | Any two from:<br>• `Credits` is a local variable // is declared inside the procedure `EndTrip`<br>• ... so it can only be used inside `EndTrip` // it cannot be used in the main program<br>• ... it no longer exists / its memory is freed once the procedure has ended (line 06) | **2** | The second and third points need `Credits` to be identified as local (or declared in the procedure). "Credits has no value" alone is too vague. |

| QUESTION | ANSWER | MARKS | GUIDANCE |
|---|---|---|---|
| **2(e)**<br>0478 8.1.6(c) | Any two from:<br>• it is changed inside the procedure and also output by the main program (line 13) // it is needed in more than one part of the algorithm<br>• it must keep its value between calls to `EndTrip` // it adds up the minutes of every trip<br>• a local variable would be lost / reset each time the procedure ends // the main program could not access a local variable | **2** | At least one point must refer to this algorithm for 2 marks; "global variables can be used anywhere" with no link to the algorithm = max 1. |
| **3(a)**<br>0478 8.1.6(c) | One mark for each reason and one mark for its explanation, max two reasons:<br>• a local variable can only be changed inside its own procedure/function ...<br>• ... so code written by other programmers cannot accidentally change its value<br>• the same identifier can be used in different procedures/functions ...<br>• ... without the variables clashing // programmers do not need to check which identifiers others have used<br>• each procedure/function is self-contained ...<br>• ... so it can be tested / debugged on its own // can be reused in another program<br>• local variables only exist while the procedure/function runs ...<br>• ... so their memory is freed when it ends // more efficient use of memory | **4** | An explanation only scores with its reason. The same reason given twice scores once. "Faster" or "more secure" alone: no mark. |
| **4(a)**<br>0478 8.1.6(b), (c) | One mark per bullet point. Max 5:<br>• `PROCEDURE RecordGroup(GroupSize : INTEGER)` and `ENDPROCEDURE`<br>• local variable declared inside the procedure, e.g. `DECLARE Books : INTEGER`<br>• number of guidebooks calculated as `DIV(GroupSize, 6)` // `GroupSize DIV 6` and stored in the local variable<br>• `TotalVisitors ← TotalVisitors + GroupSize`<br>• `OUTPUT` with a message and the local variable<br>Example:<br>`PROCEDURE RecordGroup(GroupSize : INTEGER)`<br>`  DECLARE Books : INTEGER`<br>`  Books ← DIV(GroupSize, 6)`<br>`  TotalVisitors ← TotalVisitors + GroupSize`<br>`  OUTPUT "Free guidebooks: ", Books`<br>`ENDPROCEDURE` | **5** | Accept any meaningful identifier for the local variable. `GroupSize / 6` alone, or `ROUND(GroupSize / 6, 0)` (which can round up, e.g. 11 people gives 2 instead of 1), does not get the calculation mark. Accept a correct count using a loop. Declaring `TotalVisitors` inside the procedure, or setting it to 0 there, loses the `TotalVisitors` mark. A `FUNCTION` header in place of `PROCEDURE` loses the first mark. Program code (not pseudocode): no marks. |

| | | | |
|---|---|---|---|
| **4(b)**<br>0478 8.1.6(c) | 33 | **1** | Working: 0 + 20 + 13 = 33. |

**Total: 22 marks**