MODULE M PROGRAMMING · M1

# Variables, constants, input and output: mark scheme

22 marks · OCR J277 spec 2.2.1 (the use of variables, constants, operators, inputs, outputs and assignments; sequence), with 2.1.2 (trace tables, identifying errors)

**How to use.** 1 mark per bullet up to the Max shown. **/** separates alternative words; **//** separates alternative answers for the same mark. Words in (brackets) are not needed for the mark. **...** at the start of a bullet means it depends on the bullet before it. 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. Penalise a repeated syntax slip once, then follow through.

| QUESTION | ANSWER | MARKS | GUIDANCE |
|---|---|---|---|
| **1**<br>J277 2.2.1 | Its value can be changed while the program is running: Variable<br>Its value stays the same while the program is running: Constant<br>It has an identifier (a name) that the program uses to refer to its value: Both | **3** | 1 mark per correct row. A row with more than one tick scores 0 for that row. |
| **2(a)**<br>J277 2.2.1 | `RATE` | **1** | Accept `const RATE = 4` // line 01. Do not accept "4" alone (that is the value, not the constant). |
| **2(b)**<br>J277 2.2.1 | `name` // `hours` // `cost` | **1** | Any one. If more than one is given, all must be variables. Do not accept `RATE`, `input`, `print` or `int`. |
| **2(c)**<br>J277 2.2.1 | 12 | **1** | Accept 12.0. Do not accept `cost` or `3 * 4`. |
| **2(d)**<br>J277 2.2.1 | If the hourly rate changes, it only needs to be changed in one place / on line 01 // the rate is easier to update<br>The name `RATE` makes it clear what the number means // the code is easier to read / understand<br>The value cannot be changed (by mistake) while the program is running<br>Both lines are certain to use the same value // avoids typing a different number on one of the lines | **2** | Max 2. Credit any other valid reason. Do not accept "it is faster", "it uses less memory" or "it is more secure". "Easier to change" is accepted only as updating the value in the code, not as changing it while the program runs. |

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>3(a)<br>J277 2.2.1<br>J277 2.1.2</td><td>
<table>
<tr><th>LINE</th><th>X</th><th>Y</th><th>OUTPUT</th></tr>
<tr><td>01</td><td>6</td><td></td><td></td></tr>
<tr><td>02</td><td></td><td>4</td><td></td></tr>
<tr><td>03</td><td>10</td><td></td><td></td></tr>
<tr><td>04</td><td></td><td>20</td><td></td></tr>
<tr><td>05</td><td>3</td><td></td><td></td></tr>
<tr><td>06</td><td></td><td></td><td>3</td></tr>
<tr><td>07</td><td></td><td></td><td>20</td></tr>
</table>
x: 6, 10, 3 (in this order)<br>y: 4, 20 (in this order)<br>Output: 3, 20 (in this order)
</td><td>3</td><td>1 mark per column. Line numbers are not marked. Ignore values repeated on later rows. Do not award the Output mark if extra values are output or if outputs are written as <code>print(x)</code> / x.</td></tr>
<tr><td>3(b)<br>J277 2.2.1</td><td>The lines run in sequence // one after another, in the order they are written // line 05 runs after line 03 (and before line 06)<br>Line 05 assigns 3 to x, which overwrites / replaces the 10 // a variable holds only one value at a time</td><td>2</td><td>Do not accept “x was changed” alone for the second mark: the answer must show that the old value is replaced. Do not accept “10 is stored as well” or “x is a constant”.</td></tr>
<tr><td>4<br>J277 2.2.1<br>J277 2.1.2</td><td>Line 03<br>… <code>pitchesLeft = PITCHES - booked</code><br>Line 04<br>… <code>print("Pitches left")</code></td><td>4</td><td>Each correction mark depends on the matching line number. Line 03: the error is that a constant cannot be changed / assigned a new value. Allow <code>pitchesLeft = 25 - booked</code>. Do not credit changing line 01 to make <code>PITCHES</code> a variable: <code>pitchesLeft</code> would still never be assigned. Line 04: the text must be in quotes; allow single quotes; capitals ignored. Do not credit line 05 as an error.</td></tr>
</table>

| QUESTION | ANSWER | MARKS | GUIDANCE |
|---|---|---|---|
| **5**<br>J277 2.2.1 | Constant declared with a sensible name and the value 0.15 (e.g. `const PRICE = 0.15`)<br>Number of pages input ...<br>... and stored in a variable<br>Total cost calculated as the number of pages multiplied by the constant (by name)<br>Total cost output with a message (e.g. `"Total cost: £"`)<br>**Example answer (ERL):**<br>`const PRICE = 0.15`<br>`pages = int(input("Enter the number of pages"))`<br>`total = pages * PRICE`<br>`print("Total cost: £" + str(total))` | **5** | 1 mark per bullet. Do not penalise a missing `int()` / `str()` cast (casting is 2.2.2). Allow the prompt in `input(...)` or in a `print` before it; a prompt is not required for the input marks. Calculation mark: do not award if `0.15` is typed in the calculation instead of the constant. The calculation may be done inside the output statement. Output mark: the message and the total must both be output; allow + or a comma between them. Do not accept `input` used as a variable name, or maths symbols such as × for multiply. In Python, accept a constant written as a normal variable in capitals (e.g. `PRICE = 0.15`). |

**Total: 22 marks**