MODULE O ROBUST PROGRAMS · O6

Syntax errors vs logic errors: mark scheme

22 marks · OCR J277 specification 2.3.2 (identify syntax and logic errors; syntax errors break the grammatical rules of the language and stop it from being run/translated; logic errors produce unexpected output; the purpose of testing) and 2.1.2 (identify syntax/logic errors in code and suggest fixes; trace tables)

How to use. 1 mark per bullet up to the maximum shown. **/** separates alternative words or answers. Words in (brackets) are not needed for the mark. Corrected code may be in OCR Exam Reference Language or any high-level language: mark the logic, and do not penalise minor syntax slips in the student's own code (e.g. a missing `then` or colon) unless the slip is the error the question asks about. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.3.2	Max 2 from: An error that breaks the (grammatical) rules of the programming language / code not written in the correct form for the language It stops the program from being run / translated / compiled / interpreted The translator / IDE reports it (with an error message / line number) An example, e.g. a misspelt keyword / missing bracket / missing quote mark	**2**	The example mark is only given alongside one of the first three points (an example on its own scores 0). Do not accept "an error in the syntax" (repeats the question) or "a spelling mistake" on its own. Do not accept "the program crashes while running".
1(b) J277 2.3.2	Max 2 from: An error that produces unexpected / incorrect output / results / the program does not do what the programmer intended The program (still) runs / is translated / does not break the rules of the language No error message is given / the translator does not detect it An example, e.g. > used instead of >= / a loop that runs the wrong number of times / the wrong variable used	**2**	The example mark is only given alongside one of the other points. Do not accept "an error in the logic" (repeats the question) or "the program does not work" / "the program crashes".
2(a) J277 2.3.2, 2.1.2	1 mark per correct row: `print("Card full"`: Syntax error `discount = price * 0.02`: Logic error `if stamps > 10 then`: Logic error `endfi`: Syntax error	**4**	A row with both boxes ticked scores 0 for that row. Row 2 works out 2%, not 20%. Row 3 leaves out exactly 10 stamps. Both run without an error message.
3(a) J277 2.3.2	Syntax (error)	**1**	

QUESTION	ANSWER	MARKS	GUIDANCE
3(b) J277 2.3.2	Logic (error)	**1**	Do not accept "runtime error": the program ran to the end and gave a result.
3(c) J277 2.3.2, 2.1.2	Max 2 from: Test the program with test data whose correct / expected result is known (e.g. 2 hours should give £16) Compare the actual output with the expected output Use a trace table / dry run to follow the values of the variables line by line … to find the line where a value first goes wrong Use IDE debugging tools, e.g. breakpoints / stepping through the code / watching variable values Read through / check the calculation (e.g. the hours used, the multiplication by 8)	**2**	Do not accept "the translator / error message shows the line": a logic error gives no error message. Do not accept "test it" on its own without saying what is done with the result.
4(a) J277 2.1.2, 2.3.2	Line 06 (identified) … `print("Charge: " + str(charge) + " pounds")` / closing bracket added at the end	**2**	The correction mark may be given if the line number is missing but the rewritten line makes clear which line is meant. Allow a description, e.g. "add a) at the end of line 06".
4(b) J277 2.1.2, 2.3.2	Line 03 (identified) … `if charge < 6 then`	**2**	Allow `charge <= 6` / `6 > charge` / `kg < 3` / `kg <= 3` / `kg <= 2` (all give the right charge for whole kilograms). Do not allow ≤ or =<. Do not credit changes to lines 02 or 04 (they do not fix the error).
4(c) J277 2.1.2	Charge: 6 pounds	**1**	Allow 6 on its own. Ignore quotation marks. Do not accept 10 (that is the output of the fully corrected program).

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr>
<td>5(a)
J277 2.1.2</td>
<td>i: 1, 2, 3 AND reading: 12, 9, 15 (in this order)
total: 0, 1, 3, 6
average: 2 AND Output: 2 (once, at the end)
<table>
<tr><th>I</th><th>READING</th><th>TOTAL</th><th>AVERAGE</th><th>OUTPUT</th></tr>
<tr><td></td><td></td><td>0</td><td></td><td></td></tr>
<tr><td>1</td><td>12</td><td>1</td><td></td><td></td></tr>
<tr><td>2</td><td>9</td><td>3</td><td></td><td></td></tr>
<tr><td>3</td><td>15</td><td>6</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>2</td><td>2</td></tr>
</table>
</td>
<td>3</td>
<td>Values may be repeated on later rows. Allow 2.0 for average and Output.
Allow the starting 0 for total to be missing if 1, 3, 6 are correct.
Do not give the third mark for 12 (the intended average): the trace must follow the code as written.</td>
</tr>
<tr>
<td>5(b)
J277 2.1.2, 2.3.2</td>
<td>Line 04 (identified)
… <code>total = total + reading</code></td>
<td>2</td>
<td>The correction mark may be given if the line number is missing but the rewritten line makes clear which line is meant.
Allow <code>total += reading</code>.
Do not allow <code>total =+ reading</code>.</td>
</tr>
</table>

Total: 22 marks