MODULE P LANGUAGES AND IDES · P4

IDE tools and how they help a programmer: mark scheme

20 marks · OCR J277 spec 2.5.2 (common tools and facilities in an IDE: editors, error diagnostics, run-time environment, translators; how each can be used to help a programmer develop a program) and 2.5.1 (characteristics of a compiler and an interpreter)

How to use. 1 mark per bullet up to the maximum 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 builds on the bullet before. Credit any other valid answer that shows the same understanding. A description must add more than repeating the name of the tool or feature (e.g. "line numbers number the lines" is NE).

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.5.2	1 mark for each correct row: Lets the programmer type in the code and change it: Editor Converts the high-level code into machine code: Translator Displays the fare that the program outputs while it is tested: Run-time environment Displays a message giving the line number of a mistake and what it is: Error diagnostics Automatically indents the lines inside an `if` statement: Editor	**5**	2+ ticks in a row = 0 marks for that row. Row 4: the translator may detect the error, but the tool that reports / displays it is error diagnostics.
2(a) J277 2.5.2	1 mark each to max 2: Error messages give a line number (e.g. Line 07) ... ... so the programmer can find / go straight to the line with the error (quickly) The programmer can refer to / find a particular line (e.g. when stepping through / setting a breakpoint / explaining the code to someone else) Easier to see / keep track of where you are in a long program	**2**	"Numbers each line" alone is NE (repeats the feature). "Helps find errors" alone is NE: needs the link to locating the line given in an error message or a specific line.
2(b) J277 2.5.2	1 mark each to max 2: Makes the code easier to read / understand Keywords stand out from identifiers / variables / strings // the structure (e.g. where an `if` starts) is easier to see A misspelt keyword is not coloured / shows up ... ... so (syntax) errors can be spotted (and corrected) before the program is run / more quickly	**2**	"Keywords are coloured" alone is NE (repeats the feature). "Looks nicer" = 0.

QUESTION	ANSWER	MARKS	GUIDANCE
3(a) J277 2.5.2	1 mark each to max 2: Shows the line number / location of the error (line 07) ... ... so the programmer knows where to look // can go straight to the line Describes / explains the error / what is wrong (an `endif` is missing) ... ... so the programmer knows what to change / how to fix it (add `endif` before the `print` line) (Some IDEs) suggest a fix Reports the error before / without the program having to be run // reports all the syntax errors found	**2**	"Finds errors" / "tells you there is an error" alone is NE. Answers must go beyond saying an error exists.
4(a) J277 2.5.2	1 mark each to max 4: Set a breakpoint (e.g. at line 05) ... ... the program stops / pauses at that line (so the values can be checked) Step through the program / execute the code one line at a time ... ... to see which lines run / the path taken through the code // to see that line 06 is skipped / not executed when age is 65 Use a variable watch / watch window ... ... to see the value of age / `fare` as each line runs // to see that `fare` stays at 12.0 Run the program in the run-time environment with test data (e.g. 64, 65, 66) and compare the output with the expected result ... ... to narrow down the boundary where it goes wrong (65 gives the full fare, 66 gives half) Identify that the condition on line 05 is the cause // > should be >=	**4**	Max 3 if no tool is named (breakpoint, stepping, watch, debugger, trace, run-time environment). Accept "use the debugger to ..." with a valid description. Do not accept "use the error diagnostics to find the error message": this logic error produces no error message. Allow sensible references to AI assistants that explain code, with a valid description.
5(a) J277 2.5.2	1 mark each to max 2: Lets the program be run / executed inside the IDE (without leaving it / without another program) Shows the output of the program (e.g. the fare) // an output window / console ... ... so the output can be checked / compared against the expected result The programmer can enter test data (e.g. an age) while the program runs Shows (run-time) errors that happen while the program runs // lets the programmer see where the program crashes	**2**	"Runs the program" and "shows the output" are separate marks. "Tests the program" alone is NE.

QUESTION	ANSWER	MARKS	GUIDANCE
5(b) J277 2.5.1, 2.5.2	1 mark for the benefit, 1 mark for the expansion, e.g.: Translates and runs the code one line at a time / line by line ... ... (so) it stops at the line where an error is found // the error can be located easily (and fixed) *or* No need to compile / translate the whole program first ... ... so changes can be tested quickly / straight away // part of the program can be run even if later lines are unfinished	**2**	Mark as a linked pair: the expansion must follow the benefit given. Do not accept "runs faster" (interpreted code runs more slowly than compiled code) or "produces an executable" (that is a compiler).
5(c) J277 2.5.1	Compiler	**1**	Correct answer only. Do not accept "assembler" (not on J277) or "translator" alone.

Total: 20 marks