MODULE P LANGUAGES AND IDES · P1

High-level and low-level languages: mark scheme

20 marks · Cambridge 0478 syllabus 4.2.1 (explain what is meant by a high-level language and a low-level language, including the advantages and disadvantages of each: ease of reading and writing code, ease of debugging code, machine independence, direct manipulation of hardware)

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 it builds on the bullet before it. Credit any other valid answer that shows the same understanding. In questions 3 and 4(a), answers should apply to the scenario given.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 4.2.1	Any two from: Uses English-like words / keywords // is close to human language Is easy for a human / programmer to read / write / understand Must be translated (by a compiler / interpreter) into machine code before it can be run // the CPU cannot execute it directly One statement / line can carry out many actions // is translated into many machine code instructions Is machine independent // can be used on different types of processor / computer Cannot directly manipulate the hardware // the programmer does not need to know about the hardware	**2**	max 2. Do not accept "a language that is easy" or "an advanced language" alone.
1(b) 0478 4.2.1	Any two from: Is close to the language / instructions of the processor // close to machine code Is machine code (binary) // assembly language (mnemonics) Is hard for a human / programmer to read / write / understand One instruction / line carries out a single action // each instruction matches one machine code instruction Is machine dependent // only runs on the type of processor it was written for Can directly manipulate the hardware // can directly access specific memory locations / registers	**2**	max 2. Machine code and assembly language given together score one mark. Do not accept "a language that is hard" or "a basic language" alone.
2(a) 0478 4.2.1	A	**1**	More than one extract given scores 0.

QUESTION	ANSWER	MARKS	GUIDANCE

2(b)
0478 4.2.1
	Any two from:
	It uses English-like keywords (e.g. WHILE, OUTPUT) // extract C is in binary / 1s and 0s
	It uses meaningful names / identifiers for data (e.g. Count) // extract C does not show what the data is / uses memory addresses
	It uses familiar mathematical symbols (e.g. <, +)
	Its structure (loop / indentation) is easy to see // the purpose of each line is clear
	Each statement does more work // fewer statements are needed for a task // the programmer does not need to write each machine code instruction
	The programmer does not need to know the processor's instructions / the binary codes
		2	max 2. Two answers with the same idea score once. Do not accept "it is easier to understand" with no reason (repeats the question).

3(a) 0478 4.2.1	Two advantages, one mark for each advantage and one mark for each linked expansion. Any four from: Can directly manipulate the hardware … … so the program can read the (position) sensors / send signals to the motors directly Can directly access specific memory locations / registers … … e.g. where the sensor readings are stored / where the motor signals are sent The code can run faster / more efficiently … … so the arm reacts quickly / in real time to sensor readings // so the arm stops / moves at the right moment The code can use less memory … … so it fits in the limited memory of the arm's processor The code can be written / optimised for the arm's specific processor … … so it uses the processor's own instructions	**4**	max 4, max 2 per advantage. An expansion mark must be linked to its advantage. Answers not linked to the robotic arm: max 3. Do not accept "faster" alone (must say the code runs faster). Do not accept "easier to write / debug" (wrong).
3(b) 0478 4.2.1	One mark for the disadvantage and one mark for a linked description. Any two from: The code is hard to read / write / understand … … so the section takes longer to write // mistakes are more likely // programmers need detailed knowledge of the arm's processor / hardware The code is hard to debug … … so errors take longer to find / correct The code is machine dependent // only runs on this type of processor … … so if the arm's processor is replaced with a different type, the section must be rewritten Many instructions are needed for each task … … so the code is long // it takes longer to write	**2**	max 2. Only one disadvantage is needed; if more are given, mark the best one. Do not accept "runs slower" or "uses more memory" (wrong).

QUESTION	ANSWER	MARKS	GUIDANCE
4(a) 0478 4.2.1	Any two from: The program is not written for one type of processor // can be run on different types of processor (after it is translated) So the same program can be used on the office (desktop) computers and on the tablets (in the vans) The program only needs to be written once // no need to write a separate version for each type of device / processor … which saves time / cost for the company // makes the program easier to update on all the devices	**2**	max 2. For 2 marks the answer must refer to the different devices / processors in the scenario. Do not accept "it works on any computer" with no explanation.
4(b) 0478 4.2.1	Any one from: Easier to read / write / understand (English-like) Easier to debug // errors are easier to find and correct Fewer lines of code are needed // one statement carries out many actions // quicker to write Easier to maintain / update // other programmers can understand it The programmer does not need to know about the processor / hardware	**1**	max 1. Do not credit machine independence / portability again. Do not accept "faster" or "easier" alone. Do not accept "runs faster" (wrong).
4(c) 0478 4.2.1	One mark for the disadvantage and one mark for a linked explanation. Any two from: Cannot directly manipulate the hardware / access specific memory locations … … so it is not suitable for programs that must control devices directly (e.g. device drivers / embedded control) Must be translated (into machine code) before it can be run … … so a translator (compiler / interpreter) is needed // translation takes extra time The translated code may be less efficient … … so it may run more slowly / use more memory than code written in a low-level language	**2**	max 2. Only one disadvantage is needed. Do not accept "it is slow" alone (must compare with low-level code). Do not accept "harder to read" or "machine dependent" (wrong way round).

QUESTION	ANSWER	MARKS	GUIDANCE
5(a) 0478 4.2.1	One mark for a feature and one mark for a linked explanation. Any two from: High-level code uses English-like keywords / meaningful identifiers // low-level code is in binary / mnemonics / uses memory addresses … … so the programmer can follow the logic / see what each line does // can more easily spot the line that is wrong High-level programs have fewer lines (one statement does many actions) … … so there is less code to check / trace High-level code uses clear structures such as loops and selection (not jumps to memory addresses) … … so the order the program runs in is easier to follow	**2**	max 2. Do not accept "it is easier to debug" alone (repeats the question).

Total: 20 marks