

THEME B: COMPUTATIONAL THINKING · B2.1

Programming fundamentals — Mark scheme

34 marks · covers B2.1.1–B2.1.4

How to mark. Each creditworthy point is on its own line and ends with a semicolon (;). A line beginning with ... is an *expansion* of the line above it, so the pair is worth the marks shown. A slash (/) separates acceptable alternative wordings. Words in (brackets) are not needed for the mark. Mark positively: credit what the student has got right rather than penalising what is missing, and accept any valid alternative not listed here. Do not award more than the maximum for a part. Where a later part uses a wrong answer from an earlier part correctly, award follow-through marks (**FT**). **HL** marks parts that assess higher level only content.

Q	ANSWER	MARKS																		
1(a) B2.1.1	Award [3] max Award [1] for each. Name: string; Entry fee paid: Boolean; Score to one decimal place: decimal (float / double / real); Accept char only if the candidate makes clear it is a single character, which is not appropriate for a name. Do not accept integer for the score.	3																		
1(b) B2.1.1	Award [3] max Award [1] per point up to [3] max. The contrast must be clear. A global variable is declared outside any single method or function and can be accessed throughout the program, whereas a local variable is declared inside a method or block and can only be used there; A local variable exists only while its method is running and is destroyed when the method ends, whereas a global variable exists for the life of the program; Two methods can each have a local variable of the same name without interfering, because their scopes are separate; Globals are generally avoided because any part of the program can change them, which makes faults harder to trace;	3																		
2(a) B2.1.1	Award [4] max Award [1] for each fully correct row after the seeded row. Award FT for rows that follow correctly from an earlier error. <table border="1"> <thead> <tr> <th>i</th><th>i IS EVEN?</th><th>Y</th></tr> </thead> <tbody> <tr> <td></td><td></td><td>0</td></tr> <tr> <td>1</td><td>false</td><td>4</td></tr> <tr> <td>2</td><td>true</td><td>6</td></tr> <tr> <td>3</td><td>false</td><td>10</td></tr> <tr> <td>4</td><td>true</td><td>14</td></tr> </tbody> </table> Accept 0/1, yes/no or T/F in the condition column. The key logic is that odd values of i add x (4) while even values add i itself.	i	i IS EVEN?	Y			0	1	false	4	2	true	6	3	false	10	4	true	14	4
i	i IS EVEN?	Y																		
		0																		
1	false	4																		
2	true	6																		
3	false	10																		
4	true	14																		
2(b)	Award [2] max	2																		

Q	ANSWER	MARKS
B2.1.1	Award [2] for 14. Award [1] for a value that follows correctly (FT) from an incorrect trace table. 14;	
3(a) B2.1.2	Award [4] max Award [1] for identifying the correct portion of the string; [1] for a correct substring or split operation; [1] for storing the result; [1] for printing it. Accept either language. JAVA <pre>String reference = "IB-CS-2027-HL"; String year = reference.substring(6, 10); System.out.println(year);</pre> PYTHON <pre>reference = "IB-CS-2027-HL" year = reference[6:10] print(year)</pre> Both print 2027 . Also accept splitting on the hyphen, for example <code>reference.split("-")[2]</code> in Java or <code>reference.split(" ")[2]</code> in Python, which is more robust. Java's <code>substring(6, 10)</code> and Python's <code>[6:10]</code> both take characters 6 to 9. Do not penalise an off-by-one slip more than once.	4
3(b) B2.1.2	Award [2] max Award [1] for the correct method and [1] for assigning it to a new variable. JAVA <pre>String lower = reference.toLowerCase();</pre> PYTHON <pre>lower = reference.lower()</pre>	2
4(a) B2.1.3	Award [2] max Award [1] per point up to [2] max. Any two. Unexpected input: the user types letters or symbols instead of digits, so the conversion to a number fails; Unexpected input: the user enters nothing at all, or a negative or unrealistic age; Logic error: the age is used in a calculation that assumes a sensible range; Resource unavailability: the input stream or console is not available;	2
4(b) B2.1.3	Award [4] max Award [1] for reading the input; [1] for attempting the conversion inside a try block; [1] for catching the correct exception type; [1] for a sensible response that lets the program continue. Accept either language. JAVA <pre>Scanner sc = new Scanner(System.in); System.out.print("Enter your age: "); String entered = sc.nextLine(); try { int age = Integer.parseInt(entered);</pre>	4

Q	ANSWER	MARKS
	<pre> System.out.println("Age recorded: " + age); } catch (NumberFormatException e) { System.out.println("Please enter a whole number."); } </pre> PYTHON <pre> entered = input("Enter your age: ") try: age = int(entered) print("Age recorded:", age) except ValueError: print("Please enter a whole number.") </pre> Accept a bare catch (Exception e) or except: for the third mark, but not for full marks, since catching everything hides unrelated faults. Accept a loop that keeps asking until the input is valid. Do not award the fourth mark if the handler simply prints the error and then crashes anyway.	
5(a) B2.1.3	Award [2] max Award [1] per point up to [2] max. The finally block runs whether or not an exception was raised, and whether or not it was caught; It is used for clean-up that must happen either way, such as closing the results file or releasing a resource; ...(so) the file is not left open and possibly corrupted when an error occurs part-way through writing;	2
5(b) B2.1.3	Award [3] max Award [1] per point up to [3] max. The program can recover and carry on rather than stopping, so the user does not lose the work already done; The user can be shown a clear message explaining what to do, instead of a technical error trace they cannot act on; Resources such as open files can be closed properly before the program responds; The problem can be logged for the developer while the program keeps running; An unhandled exception may leave data half-written and inconsistent; Do not credit an answer that suggests catching an exception and then ignoring it silently, which hides faults rather than handling them.	3
6(a) B2.1.4	Award [2] max Award [1] per point up to [2] max. A breakpoint pauses execution at a chosen line so the program stops before that line runs; While paused, the student can inspect the current value of every variable; They can then step through the following lines one at a time and watch where a value first becomes wrong; ...(so) the fault is narrowed down to a specific line rather than searched for across the whole program;	2
6(b) B2.1.4	Award [3] max Award up to [2] for the advantage and up to [2] for the disadvantage, [3] max. Both sides are required for [3].	3

Q	ANSWER	MARKS
	Advantage Print statements need no special tools and work in any environment; ...(so) the student can see a sequence of values over many iterations at once, which is quicker than stepping through a long loop; Disadvantage They change the source code, and forgotten prints can be left in the finished program; ...(and) they only show what the student thought to print, so a wrong value elsewhere stays invisible; Re-running is needed after every change, which is slow compared with inspecting state at a breakpoint;	

Total for this paper: 34 marks