

THEME B: COMPUTATIONAL THINKING · B2.5

File processing — Mark scheme

33 marks · covers B2.5.1

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) B2.5.1	Award [3] max Award [1] for each. Read; Write; Append; Accept the mode letters r, w and a.	3
1(b) B2.5.1	Award [3] max Award [1] per point up to [3] max. Opening a file in write mode empties it, discarding whatever it already contained; ... (so) all four existing member records would be lost as soon as the file was opened; Only the newly written record would remain in the file; Append mode instead positions the write at the end of the existing content, leaving it intact; The loss is not recoverable from the file itself, so a backup would be needed;	3
2(a) B2.5.1	Award [5] max Award [1] for opening the file in read mode; [1] for a loop that reads each line in turn; [1] for splitting the line on the comma; [1] for printing the first field only; [1] for closing the file (or using a construct that closes it automatically). Accept either language. JAVA <pre>BufferedReader reader = new BufferedReader(new FileReader("members.txt")); String line = reader.readLine(); while (line != null) { String[] parts = line.split(","); System.out.println(parts[0]); line = reader.readLine(); } reader.close();</pre> PYTHON <pre>file = open("members.txt", "r") for line in file: parts = line.strip().split(",")</pre>	5

Q	ANSWER	MARKS
	<pre>print(parts[0]) file.close()</pre> Accept a Java Scanner with <code>hasNextLine()</code> / <code>nextLine()</code>. Accept Python's <code>with open(...)</code> as <code>file:</code>, which closes the file automatically and earns the fifth mark. Award [4] max if the whole file is read at once rather than line by line, since the question asks for one line at a time.	
2(b) B2.5.1	Award [2] max Award [1] for the point and [1] for the expansion. Any one reason. Closing flushes any buffered data to disk; ... (so) data written but not yet saved is not lost when the program ends; Closing releases the file handle back to the operating system; ... (so) the resource is not held unnecessarily and other programs can access the file; An open file may be locked; ... (so) leaving it open can prevent the file being edited or deleted elsewhere;	2
3(a) B2.5.1	Award [4] max Award [1] for opening in append mode; [1] for the correct record in the same comma-separated format; [1] for writing a line ending so the next record starts on a new line; [1] for closing the file. JAVA <pre>FileWriter writer = new FileWriter("members.txt", true); writer.write("Sato,a.sato@example.com,2026\n"); writer.close();</pre> PYTHON <pre>file = open("members.txt", "a") file.write("Sato,a.sato@example.com,2026\n") file.close()</pre> The <code>true</code> second argument to Java's <code>FileWriter</code> is what selects append mode; without it the file is overwritten and no marks are awarded for the first point. Accept a missing newline only if the candidate states the previous line already ends with one.	4
3(b) B2.5.1	Award [2] max Award [1] for the point and [1] for the expansion. Any one advantage. Only the new record is written, rather than every existing record being read and written back; ... (so) it is much faster, and the saving grows as the file grows; The existing records are never rewritten; ... (so) there is no risk of corrupting or losing them if the program fails part-way through;	2
4 B2.5.1	Award [6] max Award [1] for opening the file in read mode; [1] for a counter initialised to 0; [1] for a loop over the lines; [1] for splitting each line into its fields; [1] for a correct comparison on the year field; [1] for closing the file and printing the total after the loop. Accept either language. JAVA <pre>BufferedReader reader = new BufferedReader(new FileReader("members.txt")); int count = 0; String line = reader.readLine(); while (line != null) {</pre>	6

Q	ANSWER	MARKS
	<pre>String[] parts = line.split(","); if (parts[2].equals("2025")) count = count + 1; line = reader.readLine(); } reader.close(); System.out.println(count);</pre> PYTHON <pre>file = open("members.txt", "r") count = 0 for line in file: parts = line.strip().split(",") if parts[2] == "2025": count = count + 1 file.close() print(count)</pre> The correct answer for the data shown is 2. Accept comparing the year as an integer after conversion. In Java, comparing strings with <code>==</code> instead of <code>.equals()</code> loses the comparison mark. Award [5] max if the total is printed inside the loop.	
5(a) B2.5.1	Award [2] max Award [1] per point up to [2] max. Any two. The file does not exist at the path given, or the name is misspelled; The program does not have permission to read or write the file; The file is already open and locked by another program; The storage device is full, so a write cannot complete; The file exists but its contents are not in the expected format;	2
5(b) B2.5.1	Award [2] max Award [1] per point up to [2] max. The attempt to open the file should be wrapped in exception handling (try/catch in Java, try/except in Python); The program should catch the file-not-found exception rather than stopping with an error; It should display a clear message telling the user which file is missing and where it was expected; It can then either create an empty file and continue, or exit cleanly, rather than crashing part-way through;	2
6 B2.5.1	Award [4] max Award [1] for opening report.txt in write mode; [1] for writing the heading line first; [1] for a loop writing one line per member; [1] for closing the file. JAVA <pre>FileWriter out = new FileWriter("report.txt"); out.write("Club members\n"); for (String m : members) { out.write(m + "\n"); } out.close();</pre> PYTHON	4

Q	ANSWER	MARKS
	<pre data-bbox="288 203 683 360">out = open("report.txt", "w") out.write("Club members\n") for m in members: out.write(m + "\n") out.close()</pre> Write mode is correct here because report.txt is being produced fresh each time. Do not award the second mark if the heading is written inside the loop.	

Total for this paper: 33 marks