MODULE M PROGRAMMING · M21

File handling: open, read, write and close: mark scheme

22 marks · OCR J277 spec 2.2.3 (the use of basic file handling operations: open, read, write, close), with 2.2.1 (iteration) and 2.1.2 (identifying errors)

How to use. 1 mark per bullet up to the Max 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 depends on the bullet before it. Code answers are marked for logic, not exact syntax: accept OCR Exam Reference Language or any high-level language, and allow any logically valid method. Penalise a repeated syntax slip once, then follow through.

QUESTION	ANSWER	MARKS	GUIDANCE
1 J277 2.2.3	Adds a line of text from the program to the end of the file: Write Releases the file once the program has finished using it: Close Connects the program to a named file so that it can be used: Open Copies the next line of the file into the program ...: Read	**4**	1 mark per correct row. A row with more than one tick scores 0 for that row.
2(a) J277 2.2.3	CONTENTS OF TICKETS.TXT T101 T102 T103 T104 T101 and T102 kept, at the start, in this order T103 then T104 added after them, in this order	**2**	`writeLine()` adds each line to the END of the file, so the existing lines are not lost. Ignore quotation marks round the codes and ignore empty boxes. Max 1 if any extra code is shown (e.g. a code repeated). An answer showing only T103 and T104 scores 0.

QUESTION	ANSWER	MARKS	GUIDANCE
2(b) J277 2.2.3	Creates the file: `newFile("archive.txt")` Opens the file and stores it in a variable: e.g. `archiveFile = open("archive.txt")`	**2**	Any sensible variable name. The opening mark needs the file name and a variable. Allow single quotes. In Python, `archiveFile = open("archive.txt", "w")` (or `"a"` / `"x"`) both creates and opens the file: award both marks. Python `open("archive.txt")` or `"r"` mode alone (read only, does not create) scores the opening mark only. The create mark must come before (with) the open.
3(a) J277 2.2.3 J277 2.2.1	15.25	**1**	Correct answer only (61 ÷ 4). Do not accept 61, 15 or 15.3. Do not accept `total / count`.
3(b) J277 2.2.3	4 // four (times)	**1**	Once for each line in the file. Do not accept 5.
3(c) J277 2.2.3 J277 2.2.1	It starts a loop // repeats lines 05 to 07 // repeats reading a line (and adding it to the total) ... until the end of the file is reached // while there are still lines left to read // so every line / reading in the file is read It checks whether the end of the file has been reached (before each line is read) // stops the program trying to read past the last line	**2**	Max 2. Credit any other valid description of line 04. Do not accept "it reads the file" alone (line 05 reads). Do not accept "it ends the file" or "it closes the file".
4 J277 2.2.3 J277 2.1.2	Line 01 ... `setFile = open("settings.txt")` Line 02 ... `volume = setFile.readLine()`	**4**	Each correction mark depends on the matching line number. Line 01: the file name must be in quotes; allow single quotes; allow Python `open("settings.txt", "r")`. Line 02: the program must read, not write; allow Python `setFile.readline()`. Do not accept `readLine` without the variable assignment. Do not credit lines 03 or 04 as errors.

QUESTION	ANSWER	MARKS	GUIDANCE

5
J277 2.2.3
J277 2.2.1
	Opens `orders.txt` (stored as a variable / file handle) before the orders are written
	Attempt at iteration (any loop)
	Order input inside the loop (so a new order is asked for each time)
	Loop stops when `stop` is entered, and `stop` is not written to the file
	Each order written to the file (e.g. `writeLine(order)`) inside the loop
	File closed after the loop has ended
	Number of orders counted (incremented once per order written) and output after the loop
	Example answer (ERL):
	`orderFile = open("orders.txt")`
	`count = 0`
	`order = input("Enter order")`
	`while order != "stop"`
	`  orderFile.writeLine(order)`
	`  count = count + 1`
	`  order = input("Enter order")`
	`endwhile`
	`orderFile.close()`
	`print("Orders saved: " + str(count))`
		6	1 mark per bullet, max 6. Accept any write or append mode in Python (`"w"` or `"a"`), since the file starts empty; accept `.write(order + "\n")` or `.write(order)` for writing. A Python `with open(...)` block closes the file: award the close mark if the writes are inside the block. Do-until / `while True` with `break` loops are fine if `stop` is not written. Do not award the stop mark if `stop` is written to the file or the loop never ends. Do not award the close mark if the file is closed inside the loop (unless it is reopened each time). Allow a message with the count, or the count alone. Do not accept `input` used as a variable name.

Total: 22 marks