MODULE O ROBUST PROGRAMS · O2

Input validation: mark scheme

22 marks · OCR J277 spec 2.3.1 (defensive design: input validation; how to deal with invalid data; practical experience of designing input validation), with 2.3.2 (logic errors), 2.2.3 (string length) and 2.2.1 (iteration)

How to use. 1 mark per bullet up to the Max shown. **/** separates alternative answers. Words in (brackets) are not needed for the mark. 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. Minor syntax slips are not penalised; a repeated slip (e.g. = for comparison) is penalised once then follow through. Capitals in keywords are ignored. Do not accept maths symbols (≥, ≤, ≠) or English ("is more than") in place of operators. Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.3.1	Checks data is sensible / follows set rules before it is accepted: True If an input passes validation, it must be correct: False A type check could reject `six` when a whole number is expected: True	**3**	1 mark per correct row. Both boxes ticked in a row: 0 for that row. Row 2 is False because validation only checks that data is reasonable / follows the rules: a valid value can still be the wrong value (e.g. 5 entered instead of 7).
2(a) J277 2.3.1	Member number: format check Number of days: range check Contact phone number: length check Name of the tool: presence check	**4**	1 mark per row. Accept a clear description of the check instead of its name (e.g. "checks it is between 1 and 14" for row 2; "checks it is not empty" for row 4). Row 2: accept "range check and type check"; "type check" alone = 0 (it does not test the limits 1 to 14). Row 3: do not accept range check; "format check" alone = 0 (the rule is only about the number of characters). Accept "pattern check" for row 1.

QUESTION	ANSWER	MARKS	GUIDANCE
2(b) J277 2.3.1	1 mark each to max 2: Reject the value / do not accept, store or use 20 (it fails the range check) Output an error message (that tells the member what is wrong, e.g. the number must be from 1 to 14) Ask the member to enter the number of days again / re-prompt ... ... and keep asking (in a loop) until a value from 1 to 14 is entered The program should not crash / should carry on running	**2**	Max 2. The fourth bullet depends on re-input being stated or clearly implied. "Validate it" / "check it" alone = 0. Do not accept "change it to 14" / "round it down" (the program must not silently change the input).
3(a) J277 2.3.1, 2.3.2	(Line) 02	**1**	Accept 2.
3(b) J277 2.3.1, 2.3.2	`Booking for 9 players`	**1**	Working: no number can be both less than 2 and greater than 6, so the `while` condition is always False; the loop body never runs and 9 is accepted. Ignore quotation marks. Do not accept `Team size must be 2 to 6` (alone or with other output).
3(c) J277 2.3.1, 2.3.2	`while players < 2 OR players > 6`	**1**	Accept any logically equivalent condition, e.g. `while players <= 1 OR players >= 7` / `while NOT(players >= 2 AND players <= 6)`. Each side must be a full comparison: `while players < 2 OR > 6` = 0. Allow FT only if 3(a) identified line 02 or the line is clearly the loop condition.
3(d) J277 2.3.1	1 mark for any one of: So the user is asked again / can enter a new value each time the number is invalid So the new value is checked again (by the loop condition) before it is accepted Without it, the value would never change, so an invalid value would make the loop repeat forever / an infinite loop	**1**	"To get an input" alone = 0 (must link to repeating / re-checking).

QUESTION	ANSWER	MARKS	GUIDANCE
4(a) J277 2.3.1, 2.2.3	73: `Code must be 4 characters` 73915: `Code must be 4 characters` 7391: `Locker set` (only)	**3**	1 mark per row. Ignore quotation marks round outputs. Do not award a row if `print(...)` is written round the output. Row 3: `Code must be 4 characters` as well as / instead of `Locker set` = 0 (the length is 4, so the `if` is False and the loop ends). Working: lengths are 2, 5 and 4.

5(a)
J277 2.3.1, 2.2.3, 2.2.1
	1 mark each to max 6:
	Inputs the username and stores it in a variable
	Presence check: tests whether the username is blank, e.g. `username == ""` / `username.length == 0` / `username.length < 1`
	Length check: tests the length correctly, e.g. `username.length > 12` (invalid) / `username.length <= 12` (valid)
	Attempt at a condition-controlled loop (`while` or `do ... until`) ...
	... that repeats while the username is blank OR too long / until it is not blank AND not too long, with the input (re-prompt) inside the loop
	Error message output only when invalid, and `Welcome` + username output once, after the loop
	Example:
	`username = input("Enter a username")`
	`while username == "" OR username.length > 12`
	`  print("Username must be 1 to 12 characters")`
	`  username = input("Enter a username")`
	`endwhile`
	`print("Welcome " + username)`
	Example with `do ... until`:
	`do`
	`  username = input("Enter a username")`
	`  if username.length < 1 OR username.length > 12 then`
	`  print("Invalid username, try again")`
	`  endif`
	`until username.length >= 1 AND username.length <= 12`
	`print("Welcome " + username)`
		6	Allow any logically valid method, including separate `if` statements with their own messages, a Boolean flag, or a Python `while True:` loop with `break`. The fifth bullet needs the two checks joined the right way for the loop used (AND joining the two invalid tests, or OR joining the two valid tests, loses this mark). A single `if` with no loop (only one re-entry): max 3 (input, presence check, length check). Accept `len(username)`. `username.length >= 12` as invalid (rejects 12 characters) loses the length-check mark. Do not accept `input` as a variable name. Do not accept flowcharts, structured English or bullet points.

Total: 22 marks