

2025 CpSc H Q9

Section: Software Design and Development

Topic: Testing

Question Summary

A function removal is designed to take an array of integers and a target value, returning a new array without that target. If the target is not found, the original array should be returned.

Worked Solution

(a)(i) Scope of a variable means the part of the program where that variable can be accessed or used.

Variables declared inside a function or loop are local to that block — they exist only while the block runs.

(a)(ii) Limiting scope is beneficial because it prevents unintended interference, makes debugging easier, and reduces memory use.

(b) At line 22, the program calls removal(numbers, deleteValue), executing the function with those values and returning the resulting array to newNums.

(c) Trace table for input 99:

Line	newList	position	index
5	[0,0,0]	—	—
6	—	0	—
7	—	—	0
9	[42,0,0]	—	—
10	—	1	—
7	—	—	1
9	[42,12,0]	—	—
10	—	2	—
7	—	—	2
9	[42,12,67]	—	—
10	—	3	—

Expected output: [42,12,67]

(d) A debugging technique such as breakpoints or watchpoints could be used to check values like newList, position, and index during execution.

(e)(i) The function does not produce the correct output because it initialises newList with one fewer element and fails to handle the case when the target is missing.

(e)(ii) To fix it, start with an empty newList and, after looping, return the original array if no deletion occurs:

```
FUNCTION removal(values, target) RETURNS ARRAY OF INTEGER  
DECLARE  
  newList INITIALLY []  
  FOR index FROM 0 TO length(values) - 1 DO IF
```

```
values[index] ≠ target THEN APPEND values[index] TO newList END IF END  
FOR IF length(newList) = length(values) THEN RETURN values ELSE RETURN  
newList END IF END FUNCTION (f) The function is not robust because it  
assumes valid input and non-empty arrays. After correction, it handles  
missing targets and typical edge cases correctly.
```

Revision Tips

- Scope = visibility of a variable.
- Always test functions with normal and edge-case data.
- Trace tables help confirm loop and conditional logic.
- Robustness = handling unexpected or invalid input gracefully.
- Test: value found at start, middle, end; not found; empty array.