1. What is the purpose of a loop structure?

A loop structure allows a set of instructions to be repeated multiple times until a certain condition is met. It's used to avoid writing repetitive code and to automate tasks that require iteration.

2. Explain the difference between a while statement and a do-while statement.

- while loop: The condition is tested before the loop body executes. If the condition is false at the start, the loop body may not execute at all.

- do-while loop: The loop body executes at least once, because the condition is tested after the loop body runs.

3. An input validation loop is a loop that checks user input for valid data. If valid data is not entered, the loop iterates until valid data is entered. In which review of this chapter did you write code for an input validation loop?

You likely wrote it in the Checkpoint or Coding Exercise section of this chapter where user input was checked (e.g., ensuring a positive number, valid grade, or age). It depends on your textbook, but typically this appears in the "Loop Structures Review" section.

4.

a) What is an infinite loop?

An infinite loop is a loop that never terminates because its condition is always true or never becomes false.

b) List two types of errors that can lead to an infinite loop.

1. Logic error – The loop condition is written incorrectly (e.g., while (x > 0) when x keeps increasing).

2. Update error – The loop control variable is never changed, so the condition never becomes false.

c) What is meant by overflow?

Overflow happens when a numeric variable exceeds its maximum (or minimum) storage limit, causing it to "wrap around" to an unexpected value (e.g., an int going beyond 2,147,483,647 in Java).

5. How many times will the do-while loop execute?

```
int x = 0;
do {
   x = x + 2;
} while (x < 120);
```

- The loop starts with x = 0 and adds 2 each time.

- It will continue while x < 120.

- When x reaches 120, the condition becomes false.
  The loop executes 60 times.

6. What initial value of x would make the loop infinite?

```
do {
   x = x + 3;
} while (x < 120);
```

If x starts at a value that never increases or doesn't affect the condition, it could run infinitely. But since x increases by 3 each time, the only way to make it infinite is if x never changes or if the condition never becomes false — for example:

- If you accidentally wrote the condition as while (x > 120); with x starting below 120, it would loop forever.

- In this correct code, there is no infinite loop unless you remove the increment.
  So, technically, no valid integer value of x here causes an infinite loop — but if you initialized x with a value less than 120 and didn't update it, that would do it.

7. Compare and contrast counters and accumulators. List two uses for each.

Feature	Counter	Accumulator
Purpose	Counts how many times something occurs	Adds up (accumulates) a running total
Typical operation	count = count + 1	sum = sum + value
Uses	Counting loops, tracking number of items	Summing values, calculating totals or averages

Uses:

- Counter: (1) count number of students; (2) count number of iterations.

- Accumulator: (1) total sales amount; (2) sum of test scores.

8. Write a for statement that sums the integers from 3 to 10, inclusive.

```
int sum = 0;
for (int i = 3; i <= 10; i++) {
   sum += i;
}
```

9. List two factors that should be considered when determining which loop structure to choose.

1. Whether the number of iterations is known in advance (use for loop if known; while or do-while if not).

2. Whether the loop body must execute at least once (use do-while if yes).

11. Consider the following assignment:

```
String x = "my string.";
```

Determine the value returned by each of the following methods:
a) x.length() → 10 (there are 10 characters, including space and period)
b) x.substring(0, 3) → "my " (characters at index 0, 1, and 2)