CRT chapter 5

1. What is the purpose of a loop structure?

A loop repeats a block of code until a certain condition is met.

2. Difference between a while statement and a do-while statement.

while: checks the condition before running. Might not run at all.

do-while: runs the code once first, then checks the condition. Always runs at least once.

3. Input validation loop explanation.

This loop keeps asking the user for input until they enter valid data.

Answer:
You wrote an input validation loop in the review where you checked the user’s input and repeated the loop until the input was valid.

4.

a) What is an infinite loop?

A loop that never ends because the condition never becomes false.

b) Two types of errors that cause infinite loops:

1. Forgetting to update the loop variable

2. Using the wrong condition (always true)

c) What is overflow?

Overflow happens when a value is too large for the variable’s data type.

CRT chapter 5

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

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

It runs 60 times.

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

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

Any starting value less than 120 makes it infinite, because x will keep getting smaller.

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

Counters:

Go up by 1

Used for counting items or loop repetitions

Counters can be used for:

1. Counting loop runs

2. Counting number of people/items

Accumulators:

Add totals (+=)

CRT chapter 5

Used for adding numbers together

Accumulators can be used for:

1. Adding test scores

2. Keeping total money or points

8. Write a for loop that sums integers from 3 to 10.

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

9. List two factors when choosing a loop structure.

1. Do you know how many times the loop runs? (use for)

2. Do you need it to run at least once? (use do-while)

11. Consider the assignment:

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

a) x.length()

There are 10 characters in "my string".

So the answer is:
10

b) x.substring(0, 3)

CRT chapter 5

This returns characters from index 0 up to 2.

"my string"
Indexes:
m = 0
y = 1
(space) = 2

So substring(0,3) =
"my " (that is: m y space)