CS 1324 Spring 2021 Homework 8 For Loops

Jordan McFadden

TOTAL POINTS

16 / 20

QUESTION 1

Question 12 pts

1.1 Question 1a 3 / 3

✓ **- 0 pts Correct**

- 0 pts Arithmetic error

- 0 pts Missing one of the initial accumulator or the initial count

- 1 pts Missing both initial count and accumulator

- 0 pts Missing one of final accumulator or count

- 1 pts Missing both final accumulator and count

- 3 pts Incorrect/ empty submission

1.2 Question 1b 3 / 3

✓ **- 0 pts Correct**

- 0 pts Arithmetic Error

- 1 pts For loop has more than 0 iteration.

- 3 pts Empty submission

1.3 Question 1c 3 / 3

✓ **- 0 pts Correct**

- 0 pts Arithmetic error

- 1 pts Missing the initial points value

- 0 pts Missing the intermediate points value

- 1 pts Missing the final points value

- 3 pts Incorrect/ empty submission

1.4 Question 1d 3 / 3

✓ **- 0 pts Correct**

- 0 pts Arithmetic Error

- 1 pts Not tracing 3 iterations.

- 1 pts Missing initial name/ index

- 1 pts Missing infinite loop statement.

- 3 pts Incorrect/ empty submission.

QUESTION 2

Question 2 8 pts

2.1 Question 2a 0 / 4

+ 4 pts Correct

✓ **- 1 pts Initialization of the for loop is missing or incorrect**

- 1 pts The for loop stop condition is missing or incorrect

- 1 pts The "increment/next step statement" of the for loop is missing or incorrect

- 1 pts The logic of the for loop is missing or incorrect.

+ 0 pts The "input", "SENTINEL", or "keybord" variables are missing before starting the for loop.

- 4 pts Incorrect/empty submission

The for loop should be initialized using int input= keybord.nextInt().

2.2 Question 2b 4 / 4

✓ **+ 4 pts Correct**

- 1 pts Initialization step of the while loop is missing or incorrect

- 1 pts The while loop stop condition is missing or incorrect

- 1 pts The "increment step statement" of while is missing or incorrect

- 1 pts The logic of the while is missing or incorrect

+ 0 pts The "sum", "EPSILON", or "previous" variables are missing before starting the while loop.

- 4 pts Incorrect or empty submission

Homework 8: For Loops

CS 1323/4 Spring 2021

Name: Jordan

Student ID (usually 112-XXX-XXXX or 113-XXX-XXXX): McFadden

1. (12 points; 3 points each part) Trace the for loops below in the tables to the right. If the loop is an infinite loop, trace at least three iterations and write infinite loop.

a)

```
int accumulator = 5;
for (int count = 1; count <=5; ++count)
{
        accumulator = accumulator + count * 2;
}
```

accumulator	count
5	1
7	2
11	3
17	4
25	5
35	6

b)

```
int sum = 0;
for (int count = 0; sum < 0; ++count)
{
        sum = sum – count;
}
```

sum	count
no	loop
sum	== 0

c)

```
int points = 20;
for (     ; points > 0; )
{
        points = points – 1;
        points = -1 * points – points;  // Try not to write code that looks like this!
        // But you need to understand it because sometimes others will
}
```

points
20
19
-38

d)

```
String name = “Raven”;

for (int index=0; index < name.length(); ++index)

{

        name = name + name.charAt(index);

}
```

name	index
Raven	0
RavenR	1
RavenRa	2
Infinite loop - Rains length is	
being manipulated	

2. (8 points; 4 points each)

a) Translate the while loop below into a for loop. The for loop must have a non-empty initialization, test, and increment.

```
Scanner keyboard = new Scanner(System.in);
final int SENTINEL = 0;
int input = keyboard.nextInt();
int product = 1000;
while (input > SENTINEL && product != SENTINEL)
{
       product = product / input;
       input = keyboard.nextInt();
}
keyboard.close();
```

```
Scanner keyboard = new Scanner(System.in);
int input = 0;
for (int product = 1000; product > 0; product = product/input)
{
    input = keyboard.nextInt();
    keyboard.nextLine();
}
keyboard.close();
```

b) Translate the for loop below into a while loop.

```
double sum = 0.0;
final double EPSILON = .000001;
final int LIMIT = 100;
double diff = 0.0;
for (int count = 1; diff < EPSILON && count < LIMIT; ++count)
{
       double previous = sum;
       sum = sum + 1.0 / (count * count);  // Parentheses are necessary
       diff = Math.abs(previous - sum);
}
```

```
double sum = 0.0;
final double EPSILON = .000001;
final int LIMIT = 100;
double diff = 0.0;
double previous = 0;
int count = 1;
while (count < LIMIT)
{
  if (diff < EPSILSON)
  {
    previous = sum;
    sum = sum + 1.0 / (count * count);
    diff = Math.abs(previous - sum);
    count = count + 1;
  }
  else
  {
     count = 100;
  }
}
```