Conrad Horton
CSC203 Jun 2016
Control Statements Part 2 Lab 4
20160622

Lab Part I

1. What is required to perform counter-controlled repetition?

A variable/iterator to track the number of repetitions, a target value to reach, and a loop of some kind.

2. Why should programs control counting loops with integers and not with floating-point numbers?

When manipulating floating point numbers mathematically there could be rounding issues, or your precision could be too high. With integers you do not have to worry about either of these things.

3. Explain why placing a semicolon after the header of a for statement is a logic error and not a compilation error.

You are telling the compiler that it is a complete statement, and nothing else is happening.

4. Differentiate between the while and the do...while repetition statements.

'while' checks for termination at the beginning of the loop, so there is potential you will not do anything. 'do...while' checks for termination at the end of the loop, so you will do something at least once.

5. Explain why an infinite loop can occur and how one can be prevented.

Infinite loops can occur if you do not have a terminal value or there is no way to reach your terminal value. This is avoidable if you provide a terminal value and there is a way to reach it.

Part II

For each of the given program segments, read the code and write the output for each program. [Note: Do not execute these programs on a computer.]

For the following questions, assume that the code segments are contained within the main method of a Java application.

```
1  int startingValue;
2  int terminatingValue;
3  int stepValue;
4
5  for ( int i = startingValue; i < terminatingValue; i += stepValue )
6     System.out.printf( "%d ", i );
```

1) What will be the output if the following code is placed at line 4 of the preceding code?

```
1  startingValue = 0;
2  terminatingValue = 5;
3  stepValue = 1;
```

0 1 2 3 4

2) What will be the output if the following code is placed at line 4 of the preceding code?

```
1  startingValue = -3;
2  terminatingValue = 2;
3  stepValue = 1;
```

-3 -2 -1 0 1

3) What will be the output if the following code is placed at line 4 of the preceding code?

```
1  startingValue = 6;
2  terminatingValue = 5;
3  stepValue = 1;
```

Nothing is printed

4) What will be the output if the following code is placed at line 4 of the preceding code?

```
1  startingValue = 0;
2  terminatingValue = 5;
3  stepValue = 3;
```

0 3

```
1  int startingValue;
2  int terminatingValue;
3  int stepValue;
4
5  for ( int i = startingValue; i <= terminatingValue; i += stepValue )
6  {
7     switch( i )
8     {
9        case 0:
10          System.out.print( "Hello there, " );
11          break;
12       case 1:
13          System.out.println( "What's up? " );
14          break;
15       case 2:
16          System.out.println( "How are you doing? " );
17          break;
18       case 3:
19          System.out.println( "Terrific. " );
20          break;
21       case 4:
22          System.out.println( "Beautiful day isn't it? " );
23          break;
24       case 5:
25          System.out.println( "Yes it is. " );
26          break;
27       default:
28          System.out.println( "See you later. " );
29    } // end switch
30 } // end for
```

1) What will be the output if the following code is placed at line 4 of the preceding class definition?

```
1  startingValue = 0;
2  terminatingValue = 6;
3  stepValue = 2;
```

Hello there, How are you doing?
Beautiful day isn't it?
See you later.

2) What will be the output if the following code is placed at line 4 of the preceding code?

```
1  startingValue = 0;
2  terminatingValue = 6;
3  stepValue = 3;
```

Hello there, Terrific.
See you later.

3) What will be the output if the following code is placed at line 4 of the preceding code?

```
1   startingValue = -3;
2   terminatingValue = 2;
3   stepValue = 1;
```

See you later.
See you later.
See you later.
Hello there,
What's up?
How are you doing?

4) What will be the output if the following code is placed at line 4 of the preceding code?

```
1   startingValue = -5;
2   terminatingValue = 1;
3   stepValue = 2;
```

See you later.
See you later.
See you later
What's up?

5) What will be the output if the following code is placed at line 4 of the preceding code?

```
1   startingValue = 10;
2   terminatingValue = 5;
3   stepValue = 1;
```

Nothing is printed.

6) What is output by the following code segment?

```
1  for ( int i = 0; i <= 11; i++ )
2  {
3     if ( i % 2 == 0 )
4        continue;
5
6     if ( i == 11 )
7        break;
8
9     System.out.printf( "%d ", i );
10 } // end for
```

1 3 5 7 9