CS 172 - Lab 4

Submit **Lab4.java** on Canvas.

Make sure your program compiles. Starting with this lab assignment, any program that does not compile will receive zero points.

The purpose of this lab is to give you considerable practice using for loops.

Schedule Your Time

I estimate that this lab will take about 6 hours to complete. Get out your calendar, planner, cell phone, or whatever you use to schedule your time. Schedule the times that you are going to work on this assignment. You can choose one hour blocks, two hour blocks, or one six hour block.

Read this material before you begin the lab assignment.

Java - For Loop

A for loop is one method of repeating a statement or block of statements. The for loop structure looks like this

```
for ( initialization ; condition ; increment ) {

        // body of the loop

        // contains statement(s) to be repeated

} // end loop
```

The *initialization* can be empty, or it may contain one or more statements. Typically, the initialization portion of the loop header is just one assignment statement that sets an initial value for a variable.

The *condition* must be a boolean expression. It must evaluate to either true or false.

The *increment* may be empty, or it may contain one or more statements. The increment must be a complete Java statement. Examples:

```
x = x + 1          // complete Java statement

a++                // complete Java statement


n + 1              // NOT a complete Java statement

 b * 2             // NOT a complete Java statement
```

The *body* of the loop contains one or more statements that are repeated as long as the condition is true. Once the condition becomes false, the loop stops, and the program jumps to the statement immediately after the end of the loop.

Example For Loop

```
for (int n = 1; n < 10; n = n + 1) {

      System.out.println(n);

} // end for
```

Loop Control Variable

A *loop control variable* is a variable that determines how many times the loop repeats. In the example above, the loop control variable for the loop is n.

Although n isn't a very meaningful name for a variable, it is common to see one-letter names for loop control variables. You should choose meaningful names whenever possible.

For Loop versus While Loop

Every for loop has an equivalent while loop. Taking the pieces of the for loop shown above, we can make a while loop that will do the same thing. The pieces just get rearranged:

```
initialization;
while (condition) {
   // body of the loop
  increment;
} // end loop
```

Nested For Loops

A for loop can be nested within another for loop. Nested means that one loop is placed inside the body of another loop. It's also possible for any combination of while loops and for loops to be nested.

Nested loops are very common in two-dimensional problems. Here is a short example with two nested for loops:

```
for ( int i = 1; i < 4; i++ ) {
   for ( int k = 1; k < 3; k++ ) {
         System.out.print((i * k) + " ");
   } // end k loop
   System.out.println( );
} // end i loop
```

The outer for loop (controlled by i) repeats 3 times. Every time the outer loop repeats, the inner loop (controlled by k) repeats 2 times. What is the output of this code segment?

Output
1 2
2 4
3 6

Clearing Out The Input Buffer

Important ❋

Important ❋

Important ❋

1. Input a number
2. Clear out the Input buffer
   `scan.nextLine();`
3. Input a String

The Lab Assignment

There are 7 problems in this lab. All of the code for the 7 problems goes into one Java program called Lab4.java. This program will be both an application and an applet when it is completed.

Definition: A Java *application* is a program with a *main* method. Applications can run "stand alone", without the need for another program or an html file.

Documentation and Style Requirements

- Put a program header with comments just like you did in the previous lab assignments.
- For each problem, use System.out.println to print a message with the problem number, then write the statements to solve that problem.
- Print a blank line before and a blank line after the output of each problem.
- Do the problems in order.
- COMMENT EACH PROBLEM CLEARLY! Comments before each problem's code should have the problem number and a brief description of problem. The TA should not have to hunt for the code to each problem. Here's an example of the type of comments and the print statement I want you to have:

```
// problem # 1 - print a triangle of asterisks
//
System.out.println ("This is problem # 1\n");
```

If you get stuck on a particular problem, put the comments and the print statements in your program. Then move on to the next problem.

Come back to that problem later.

Problem 1

Write a for loop to print the odd numbers from 1 to 99 (inclusive). Print one number per line.

There is no input from the user on this problem.

Problem 2

Input a number from the user (use Scanner class method nextInt). Then, write **two nested for loops** to display a triangle made of asterisks. The number the user types determines the number of lines in the triangle. For example, if the user typed 5, the triangle should look like:

```
*
**
***
****
*****
```

The minimum number that the user can enter is 1. The maximum number is 50. Put an if statement to check for this. If the user enters a number less than 1 or greater than 50, print an error message "Invalid input". Don't print any asterisks if the input is invalid.

✱ After the code for this program, clear out the input buffer.

Problem 3

Input a String from the user (use Scanner class method nextLine). Then, write a for loop that prints the characters of the String, one at a time, in reverse.

Problem 4

Write a for loop that will input 10 integer values from the user and keep track of the highest value. After the loop is over, print the highest value.

✱ After the code for this program, clear out the input buffer.

Problem 5

Input a String from the user (use Scanner class method nextLine). Then, write a for loop that counts how many of each vowel (a, e, i, o, u) are found in the String. After the loop is over, display the number of times each vowel was found. Case does not matter. 'A' and 'a' should be counted as the vowel a.

The output should be neatly aligned as shown below:

```
The string contained:
20 A's
 6 E's
 7 I's
12 O's
 0 U's
```

Remember that you can use System.out.printf to easily format numeric output.

Problem 6

Using two nested for loops, display a table of numbers and their squares from 1 to 80 in a table like this:

```
Num Square Num Square Num Square Num Square
  1      1  21    441  41   1681  61   3721
  2      4  22    484  42   1764  62   3864
.
.
.
```

There is no input from the user on this problem.

Problem 7

Using two nested for loops, display lists of all of the factors of the numbers from one to 50. The numbers should be right-aligned. The factors should be aligned as well.

Here are the first few lines of the output so you can see how the alignment should be.

```
Number      Factors
------      -------
   1         1
   2         1  2
   3         1  3
   4         1  2  4
   5         1  5
   6         1  2  3  6
   7         1  7
   8         1  2  4  8
   9         1  3  9
  10         1  2  5 10
```

There is no input from the user on this problem.

Submit Lab4.java on Canvas

(You may submit an assignment multiple times. Only your last submission will be graded.)