CSCI 241 Assignment 5 Practicing Loops

Due: Thursday, March 18, 2021 @11:59 p.m.
Note: due to the exam, you have two weeks to complete this assignment.

Objective: Practice writing 3 kinds of loops and use Java's **`printf()`** command to format output. Your BlueJ project will contain 3 different classes, one for each type of loop.

This program is worth a lot of points… but there are a lot of parts to it! You will use loops throughout the rest of the semester and it is critical to your success to understand them.

For 3 points, email your instructor the lines of code for part B of the **`WhileLoops`** program (in the email message) by 11:59 p.m. on Thursday, March 11.

Description:

Create a new BlueJ project named **`Assign5Loops`**.
Add 3 classes: **`WhileLoops`**, **`DoWhileLoops`** and **`ForLoops`**

Each program will run the type of loop in its title:
`WhileLoops`: 3 `while` loops
`DoWhileLoops`: 2 `do-while` loops
`ForLoops`: 3 `for` loops and one nested-`for` loop.

After each program description you will see an example of the output. Within the same program, between loops, you will see a line of hyphens (------). Please make sure you match the spacing, spelling and capitalization that you see in the samples of output for each program (if the same input values are entered).

`WhileLoops`

Section A: prints the word `March`, then uses a `while` loop to print 3 copies of this pattern of characters: `-\ABC/-`, then prints `2021`. Notice that you will need escape sequences to print the \ character in the pattern and the " characters that are in the printed heading.

Section B: uses a `while` loop to read in unknown number of positive integers. It will keep counts of the number of integers that are odd and even. The loop will stop when the user types a non-positive integer (zero or negative) After the loop, it will print the counts.

Section C contains 1 (or 2) **`while`** loops (yes, it can be done with one loop!)
This section will print the integer values 8 through 48, by eighths, tab distance apart (remember the `\t` character?), on one line, and the integers in the opposite sequence in a separate line of output.
Hint: If you wish to take on the challenge of using only one loop, use a Boolean variable to keep track of whether you are counting up or counting down.

Sample output for **WhileLoops** (data entered from keyboard is **bold and underlined** to distinguish it from what the program prints):

```
          ------ Section A ------
(while) Loop runs and places 3 -\ABC/- symbol patterns
        between "March" and "2021".
================================================================================
March-\ABC/--\ABC/--\ABC/-2021

--------------------------------------------------------------------------------

          ------ Section B ------
(while) loop asks for positive integers until you enter 0 or lower.
The loop will count the number of odd and even integers entered.
Both counts will be printed after the loop finishes.
================================================================================
Enter a positive integer, 0 or lower to terminate: 5
Enter an integer, 0 to terminate: 98
Enter an integer, 0 to terminate: 134
Enter an integer, 0 to terminate: 3
Enter an integer, 0 to terminate: 47
Enter an integer, 0 to terminate: 72
Enter an integer, 0 to terminate: 15
Enter an integer, 0 to terminate: 85
Enter an integer, 0 to terminate: 0
Even numbers entered: 3
Odd numbers entered: 5

--------------------------------------------------------------------------------

          ------ Section C ------
(while) loop prints 8 through 48, then 48 down to 8, by eights.
Each set of numbers appears on its own line, tab distance apart.
================================================================================
8       16      24      32      40      48
48      40      32      24      16      8
```

DoWhileLoops

Section A: after printing a heading, **do-while** loop asks the user to enter a number that is a multiple of 5. It will keep asking until the user types a number that fits the criteria. If the number entered is not a multiple of 5, it should print this error message:

```
Error: number must be a multiple of 5.
```

After a correct number is entered, the loop will end, and the number will be printed afterwards.

Section B: after printing heading, **do-while** loop asks the user to enter a number that is a between 0 and 12, inclusive (meaning that both 0 and 12 will be included as "good" answers, as well as all values between them). It will keep asking until the user types a number in that range. If the number entered is not in that range, it should print this error message:

```
Error: number must be between 0 and 12, inclusive.
```

After a correct number is entered, the loop will end, and the number will be printed afterwards.

Sample output for **DoWhileLoops** (data entered from keyboard is **bold and underlined** to distinguish it from what the program prints):

```
          ------ Section A ------
(do-while) User enters number that is a multiple of 5.
After the loop, it prints the final accepted number.
========================================================================
Enter a number that is a multiple of 5: 14
Error: number must be a multiple of 5.
Enter a number that is a multiple of 5: 63
Error: number must be a multiple of 5.
Enter a number that is a multiple of 5: 55
Multiple of 5 entered: 55

------------------------------------------------------------------------

          ------ Section B ------
(do-while) User enters number between 0 and 12, inclusive.
After the loop, it prints the final accepted number.
========================================================================
Enter a number between 0 and 12, inclusive: -8
Error: number must be between 0 and 12, inclusive.
Enter a number between 0 and 12, inclusive: 42
Error: number must be between 0 and 12, inclusive.
Enter a number between 0 and 12, inclusive: 12
The chosen number is: 12
```

ForLoops

Section A: after asking for a positive digit from the keyboard, **for** loop does repeated addition to find the sum of 1 through that number. For example, if the number entered is 4, the loop will perform one add operation each time to get the final answer of 10 (which is 1+2+3+4). It will print the sum after the loop.

Section B: after printing a heading, **for** loop will print the values from 48 down to 4, by 4's. Each of the printed numbers should be separated by underscores. *Note: print a zero (0) after the loop finishes.*

Section C: First, print the table heading as seen in the sample output. This loop will create a table of values from 10 to 100 (each representing a distance in yards) and its equivalent in meters, increasing by 10 yards with each line. For each value, calculate equivalent meters and print first yards (use 3 positions) then meters (specify 2 decimal places) in a neat table.
Note: please use this conversion factor: 1 yard = 0.9144 meters (a good constant!)

Section D: contains one nested-**for** loop.
This section will print rows of integers. All integers in a row will be the same.
The outer loop will run 10 times and the inner loop will run 5 times.
Begin with the number 9. In the outer loop, you will count down by one each time. *And…* to make it interesting (!), if the number is 3,6 or 9, print it surrounded by a caret (^) symbol, otherwise, surround it with space characters.
Outer **for**: runs 10 times and subtracts one from the counting number each time. It will print a new line when appropriate.

Inner **for**: runs 5 times. It will determine whether to print the number surrounded by spaces or carets.

Sample output for **ForLoops** (data entered from keyboard is **bold and underlined** to distinguish it from what the program prints):

```
          ------ Section A ------
Please enter a positive digit: 7
(for) Loop finds the sum of values between 1 and the entered number.
========================================================================
Sum of values 1 through 7 = 28

------------------------------------------------------------------------

          ------ Section B ------
(for) Loop prints values 48->4, separated by underscores, by 4's.
========================================================================
48_44_40_36_32_28_24_20_16_12_8_4_0

------------------------------------------------------------------------

          ------ Section C ------
(for) Loop shows range of values representing distance measurements
      from 10 to 100 yards.
Using increments of 10, calculates equivalent meters.
========================================================================
       Yards      Meters
--------------------------------
         10         9.14
         20        18.29
         30        27.43
         40        36.58
         50        45.72
         60        54.86
         70        64.01
         80        73.15
         90        82.30
        100        91.44

------------------------------------------------------------------------

          ------ Section D ------
(Nested for-loop)
Starting with the digit 9, print 10 rows of digits.
If the digit is 3 or 6 or 9, print the digit surrounded by ^ symbols.
If the digit any other number, print the digit surrounded by spaces.
The outer loop will control the row number.
The inner loop will print the same value 5 times.
========================================================================
 Rows of Digits
---------------
^9^^9^^9^^9^^9^
 8  8  8  8  8
 7  7  7  7  7
^6^^6^^6^^6^^6^
 5  5  5  5  5
 4  4  4  4  4
^3^^3^^3^^3^^3^
 2  2  2  2  2
 1  1  1  1  1
 0  0  0  0  0
```

Since the type of loop to use is clearly specified in each program, *no credit will be granted for any code that does NOT use a loop, or that uses a different type of loop.*

Notes and Hints:

1. Sometimes you will need to update more than one variable inside a loop. Don't forget to initialize all variables *before* each loop condition is checked!

2. When you use a `System.out.printf()` statement, remember to place a `'\n'` character at the end to make it print a new line. To get a tab distance, use the `'\t'` character.

Submission Requirements

1. **Electronic:** Submit the completed implementation in the lab for the **`Assign5Loops`** project through BlueJ's submission process.
2. **Canvas Copies:** Upload/Submit a copy of the source code of each program (`WhileLoops.java, DoWhileLoops.java and ForLoops.java)` to Canvas.

Grading Criteria

Your program must compile without errors to be accepted. **Note: This requirement will be strictly enforced on this and subsequent assignments!!**

All of the following criteria will be used to evaluate your program:

Email to your instructor by 11:59 p.m. on Thursday, March 11	*3 pts*
Correct submission of project	2 pts
Appropriate and complete comments	6 pts
Adherence to style and design conventions (identifier names, indentation, etc.)	3 pts
Printed lines before and after loops	4 pts
`WhileLoops`	7 pts
`DoWhileLoops`	6 pts
`ForLoops`	14 pts
Total points:	**45 pts**