CS 101 - Algorithms & Programming I

Fall 2024 - Lab 4

Due: Week of October 21, 2024

*Remember the **honor code** for your programming assignments.*
*For all labs, your solutions must conform to the CS101 style **guidelines**!*
All data and results should be stored in variables (or constants where appropriate) with meaningful names.

The objective of this lab is to learn how to use a while loop to implement automated repetition. Remember that analyzing your problems and designing them on a piece of paper before starting implementation/coding is always a best practice.

In this particular lab, only **use the while loop**, do *not* use the for or do-while loops.

0. Setup Workspace

Start VSC and open the previously created workspace named `labs_ws`. Now, under the `labs` folder, create a new folder named `lab4`.

In this lab, you are to have four Java classes/files (under `labs/lab4` folder) as described below. A fifth and sixth Java file containing the revisions should go under this folder as well. We expect you to submit a total of 6 files including the revisions, **without compressing** them. Do *not* upload other/previous lab solutions in your submission. The user inputs in the sample runs are shown in blue.

For all parts of this assignment, you may assume that the user enters valid types and number of values unless stated otherwise (e.g., a positive integer when asked for a positive integer).

1. Factorial Calculator

Write a Java program that calculates the factorial of a given positive integer using a while loop. The program should prompt the user to enter a number and calculate its factorial. The program should continue to ask for inputs until the user enters a non-positive integer.

Sample run:

```
Enter a number: 5
Factorial of 5 is 120

Enter a number: 3
Factorial of 3 is 6

Enter a number: -1
Program finished.
```

2. Sum of Digits

Write a Java program that repeatedly asks the user to input a positive integer. The program should use a while loop to calculate the sum of digits of the number. The program should stop when the user enters zero.

Sample runs:

```
Enter a positive number: 1234
Sum of digits: 10

Enter a positive number: 567
Sum of digits: 18

Enter a positive number: 0
Program finished.
```

3. Reversing a Number with Constraints

Write a program that prompts the user for a positive integer and uses a while loop to reverse the digits of the number. It should also calculate the sum of the digits simultaneously. The program should print the reversed number and the sum of digits. Continue until the user enters a number with more than 8 digits or a non-positive integer. You are to work with integers for this question and not use other types including `String`.

Sample run:

```
Enter a positive number: 12345
Reversed number: 54321
Sum of digits: 15

Enter a positive number: 543210
Reversed number: 12345
Sum of digits: 15

Enter a positive number: 987654321
Number exceeds 8 digits.
Program finished.
```

.

4. Prime Number Sum and Count

Write a Java program that calculates the sum of all prime numbers between 1 and n where n is a positive integer provided by the user. After computing the sum and the number of prime numbers found, the program should ask the user if they want to perform a similar calculation for another n. The program should use a while loop and stop when the user chooses to quit.

Sample run:

```
Enter a positive number: 10
Prime numbers: 2 3 5 7
Sum of primes: 17
Count of primes: 4

Do you want to calculate again? (yes/no): yes
Enter a positive number: 20
Prime numbers: 2 3 5 7 11 13 17 19
Sum of primes: 77
Count of primes: 8

Do you want to calculate again? (yes/no): no
Program finished.
```