Guided LAB 303.5.2 - While Loop

Lab Objective:

In this lab, we will demonstrate how to use the ***while* loop**. We will also look into the advanced Java classes that we will study in later lectures.

Learning Objective:

By the end of this lab, learners will be able to use the ***while loop*** in Java.

While *loop* Overview

A w*hile loop* will continually process a statement or given code block while its evaluation expression is true.

```
while (condition) {
    statement(s)
}
```

Example #1: Guess the Number

- Write a program that randomly generates an integer between 0 and 100, inclusive. The program should prompt the user to enter a number repeatedly until the number matches the randomly generated number.
- For each user input, the program tells the user whether the input is too low or too high.
- When the user discovers the correct answer, the program outputs a congratulatory message, and then provides the user with the opportunity to play again.

Solution: Guess the Number

Make a new class named ***GuesstheNumber***. Inside, write the code shown below.

```
import java.util.Scanner;
public class GuesstheNumber {
   public static void main(String[] args) {
       // Generate a random number to be guessed
       int number = (int) (Math.random() * 101);
       Scanner input = new Scanner(System.in);
       System.out.println("Guess a magic number between 0 and 100");
       int guess = -1;
       while (guess != number) {
           // Prompt the user to guess the number
           System.out.print("\nEnter your guess: ");
           guess = input.nextInt();
           if (guess == number)
               System.out.println("Yes, the number is " + number);
           else if (guess > number)
               System.out.println("Your guess is too high");
           else
               System.out.println("Your guess is too low");
       } // End of Loop
   }
}
```

Example #2: An Advanced Math Tool

- Write a program that generates five single-digit integer subtraction problems.
- Using a *while* loop, prompt the user to answer all five problems.
- After all of the answers are entered, report the number of the correct answers. Offer the user the opportunity to play the game again.

Solution: Make a new class named ***SubtractionQuizLoop***. Inside, write the code shown below.

```
public class SubtractionQuizLoop
public static void main(String[] args) {
   final int NUMBER_OF_QUESTIONS = 5; // Number of questions
   int correctCount = 0; // Count the number of correct answers
   int count = 0; // Count the number of questions
   long startTime = System.currentTimeMillis();
   String output = ""; // output string is initially empty
   Scanner input = new Scanner(System.in);

   while (count < NUMBER_OF_QUESTIONS) {
       // 1. Generate two random single-digit integers
       int number1 = (int) (Math.random() * 10);
       int number2 = (int) (Math.random() * 10);
       // 2. If number1 < number2, swap number1 with number2
       if (number1 < number2) {
           int temp = number1;
           number1 = number2;
           number2 = temp;
       }
       // 3. Prompt the student to answer "What is number1 - number2?"
       System.out.print(
               "What is " + number1 + " - " + number2 + "? ");
       int answer = input.nextInt();
       // 4. Grade the answer and display the result
       if (number1 - number2 == answer) {
           System.out.println("You are correct!");
           correctCount++;
       } else
```

```
            System.out.println("Your answer is wrong.\n" + number1
                     + " - " + number2 + " should be " + (number1 -
number2));
       // Increase the count
       count++;
       output += "\n" + number1 + "-" + number2 + "=" + answer +
               ((number1 - number2 == answer) ? " correct" : " wrong");
   }
}
}
```

Example #3: Controlling a Loop with a Sentinel Value

- Write a program that reads and calculates the sum of an unspecified number of integers. The input 0 signifies the end of the input.
- If data is not 0, it is added to the sum, and the next input data are read. If data is 0, the loop body is not executed, and the while loop terminates.
- If the first input read is 0, the loop body never executes, and the resulting sum is 0.

Suggested Output

```
● Enter an int value (the program exits if the input is 0): 2
● Enter an int value (the program exits if the input is 0): 3
● Enter an int value (the program exits if the input is 0): 4
● Enter an int value (the program exits if the input is 0): 0
● The sum is 9
```

Solution: Make a new class named ***SentinelValuedemo***. Inside, write the code shown below.

```
public class SentinelValuedemo{
   public static void main(String[] args) {
        // Create a Scanner
        Scanner input = new Scanner(System.in);
        // Read an initial data
        System.out.print("Enter an int value (the program exits if the input is 0): ");
        int data = input.nextInt();
        // Keep reading data until the input is 0
        int sum = 0;
        while (data != 0) {
          sum += data;
          // Read the next data
          System.out.print( "Enter an int value (the program exits if the input is 0): ");
          data = input.nextInt();
        }
        System.out.println("The sum is " + sum);
    }
```

Submission Instructions:

Include the following deliverables in your submission -

- Submit your source code using the Start Assignment button in the top-right corner of the assignment page in Canvas.

CANVAS STAFF USE ONLY: Canvas Submission Guideline:

Instructions for Canvas Assignment Creation
Assignment Name:GLAB - 303.5.2 - While Loop

Points: 100

Assignment Group: Module 303: Java SE Review (Not Graded)

Display Grade As: Complete/Incomplete

Do not count this assignment towards the final grade: Checked

Submission Types: Files Uploads

Everything else is the default.