Lab 05

Repetition Control Structures

Objective:

This lab will give you practical implementation of different types of Repetition Control Structures.

Activity Outcomes:

On completion of this lab student will be able

- Write a while loop and its variation (counter-controlled, sentinel-controlled and flag-controlled)
- Write a do-while loop

Instructor Note:

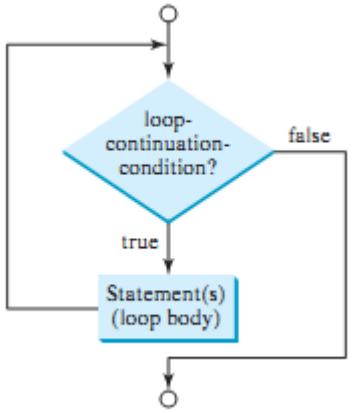
As a pre-lab activity, read Chapter 05 from the text book "Java How to Program, Deitel, P. & Deitel, H., Prentice Hall, 2019".

1) Useful Concepts

The `while` Loop: A **`while`** loop executes statements repeatedly while the condition is true.

Syntax

```
while(loop-continuation-condition){
  // Loop body
  Statement(s);
}
```



Counter-Controlled `while` Loops

Suppose you know exactly how many times certain statements need to be executed. For example, suppose you know exactly how many pieces of data (or entries) need to be read. In such cases, the **`while`** loop assumes the form of a counter-controlled **`while`** loop.

```
counter = 0; //initialize the loop control variable
while (counter < N){     //test the loop control variable
    ...
    counter++; //update the loop control variable
}
```

Sentinel-Controlled `while` Loops

Another common technique for controlling a loop is to designate a special value when reading and processing a set of values. This special input value, known as a sentinel value, signifies the end of the input. A loop that uses a sentinel value to control its execution is called a *sentinel-controlled* loop

```
input the first data item into variable //initialize the loop control
                                        //variable
while (variable != sentinel){        //test the loop control variable
        . . .
        input a data item into variable //update the loop control
        . . .                           //variable
}
```

Flag-Controlled `while` Loops

A flag controlled **`while`** loop use s a **`boolean`** variable to control the loop. Suppose **`found`** is a boolean variable. The flag-controlled **`while`** loop takes the following form

```
found = false ; //initialize the loop control variable
while (!found){ //test the loop control variable
      . . .
```

```
      if (logical expression)
      found = true ; //update the loop control variable
      . . .
}
```

The `for` Loop

The `do-while` Loop

A **`do-while`** loop is the same as a while loop except that it executes the loop body first and then checks the loop continuation condition.

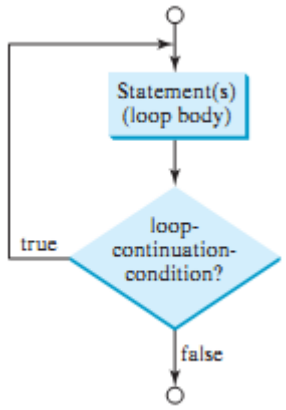
Syntax

```
do{

      // Loop body;

     Statement(s);

}while(loop-continuation-condition);
```



2) Solved Lab Activites

Sr.No	*Allocated Time*	*Level of Complexity*	*CLO Mapping*
Activity 1	*10 mins*	*Midum*	*CLO-5*
Activity 2	*10 mins*	*Midum*	*CLO-5*
Activity 3	*10 mins*	*Midum*	*CLO-5*
Activity 4	*10 mins*	*Midum*	*CLO-5*
Activity 5	*10 mins*	*Medium*	*CLO-5*

Activity 1:

This program illustrate the usage of simple while loop. It prompts the user to enter an answer for a question on addition of two single digits.

Solution:

```
import java.util.Scanner;
public class Activity1{
      public static void main(String[] args) {
            int number1 = (int)(Math.random() * 10);
            int number2 = (int)(Math.random() * 10);
            Scanner input = new Scanner(System.in);
            System.out.print("What is " + number1 + " + " + number2 +
      "? ");
            int answer = input.nextInt();
            while (number1 + number2 != answer) {
                  System.out.print("Wrong answer. Try again. What is
            "+ number1 + " + " +           number2 + "? ");
                  answer = input.nextInt();
            }
            System.out.println("You got it!");
      }
}
```

Output

```
What is 5 + 9? 12
Wrong answer. Try again. What is 5 + 9? 34
Wrong answer. Try again. What is 5 + 9? 14
You got it!
```

Activity-2:

This program illustrate the working of counter-controlled `while` loop. Suppose the input is: 8 9 2 3 90 38 56 8 23 89 7 2 (limit is defined by user).Suppose you want to add these numbers and find their average.

Solution:

```
import java.util.*;
public class Activity2{
      public static void main(String [] args){
            Scanner console = new Scanner(System.in);
            int limit;
            int number;
            int sum;
            int counter; //loop control variable
            System.out.print("Enter the number of " + "integers in the
      list: ");
            limit = console.nextInt();
```

```
            System.out.println();
            sum = 0;
            counter = 0;
            System.out.println("Enter " + limit+ " integers.");
            while (counter < limit){
                  number = console.nextInt();
                  sum = sum + number;
                  counter++;
            }
            System.out.printf("The sum of the %d "+"numbers= %d%n",
      limit, sum);
            if (counter != 0)
                  System.out.printf("The average = %d%n",(sum /
            counter));
            else
                  System.out.println("No input.");
      }
}
```

Output

```
Enter the number of integers in the list: 12
Enter 12 integers.
8 9 2 3 90 38 56 8 23 89 7 2
The sum of the 12 numbers = 335
The average = 27
```

Activity 3:

This activity demonstrate the usage of sentinel-controlled `while` loop. Suppose you want to read some positive integers and average them, but you do not have a preset number of data items in mind. Suppose you choose the number -999 to mark the end of the data.

Solution:

```
import java.util.*;
public class Activity3{
      static Scanner input = new Scanner(System.in);
      static final int SENTINEL = -999;
      public static void main(String [] args){
            int number; //variable to store the number
            int sum = 0; //variable to store the sum
            int count = 0; //variable to store the total
            System.out.println("Enter positive integers "+ "ending
      with " + SENTINEL);
            number = input.nextInt();
            while (number != SENTINEL)
```

```
            {
                  sum = sum + number;
                  count++;
                  number = input.nextInt();
            }
            System.out.printf("The sum of %d " +"numbers = %d%n",
      count, sum);
            if (count != 0)
               System.out.printf("The average = %d%n", (sum / count));
            else
                  System.out.println("No input.");
      }
}
```

Output

```
Enter positive integers ending with -999
34 23 9 45 78 0 77 8 3 5 -999
The sum of 10 numbers = 282
The average = 28
```

Activity 4:

This activity demonstrate the usage of flag-controlled `while` loop The following program randomly generates an integer greater than or equal to 0 and less than 100 . The program then prompts the user to guess the number. If the user guesses the number correctly, the program outputs an appropriate message. Otherwise, the program checks whether the guessed number is less than the random number. If the guessed number is less than the random the number generated by the program, the program outputs the message, ''Your guess is lower than the number''; otherwise, the program outputs the message, ''Your guess is higher than the number''. The program then prompts the user to enter another number. The user is prompted to guess the random number until the user enters the correct number.

Solution:

```
import java.util.*;
public class Activity4{
      static Scanner input = new Scanner(System.in);
      public static void main(String [] args){
            //declare the variables
            int num; //variable to store the random number
            int guess; //variable to store the number
            //guessed by the user
            boolean done; //boolean variable to control the loop
            num = ( int) (Math.random() * 100);
            done = false ;
            while (!done){
                  System.out.print ("Enter an integer greater"+"than or
                  equal to 0 and "+"less than 100: ");
```

```
                guess = input.nextInt();
                System.out.println();
                if (guess == num){
                      System.out.println("You guessed the "+"correct
                      number.");
                      done = true;
                } //Line 12
                else if (guess < num)
                      System.out.println("Your  guess  is  "  +"lower
                than" + "the number.\n" + "Guess again!");
                else //Line 15
                      System.out.println("Your  guess  is  "+"higher
                than" + "the number.\n"+"Guess again!");
          } //end while
     }
}
```

Output

```
Enter an integer greater than or equal to 0 and less than 100: 25
Your guess is higher than the number.
Guess again!
Enter an integer greater than or equal to 0 and less than 100: 5
Your guess is lower than the number.
Guess again!
Enter an integer greater than or equal to 0 and less than 100: 10
Your guess is higher than the number.
Guess again!
Enter an integer greater than or equal to 0 and less than 100: 8
Your guess is higher than the number.
Guess again!
Enter an integer greater than or equal to 0 and less than 100: 6
Your guess is lower than the number.
Guess again!
Enter an integer greater than or equal to 0 and less than 100: 7
You guessed the correct number.
```

Activity 5:

This activity demonstrate the working of do-while loop. Following program keep reading data until the input is 0. The program will display maximum of the numbers. Suppose the input is 2 3 4 5 0.

Solution:

```
import java.util.Scanner;
public class Activity6 {
      public static void main(String[] args) {
            Scanner input = new Scanner(System.in);
            int number, max;
            number = input.nextInt();
            max = number;
            do {
                  number = input.nextInt();
                  if (number > max)
                          max = number;
                  }while (number != 0);
           System.out.print("max is " + max+" and number " + number);
      }
}
```

Output

```
max is 5 and number 0
```

3) Graded Lab Tasks

Note: The instructor can design graded lab activities according to the level of difficult and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab.

Lab Task 1

- ***a)*** *Given two integers A and B ($A \leq B$). Print all numbers from A to B inclusively.*
- ***b)*** *Given two integers A and B. Print all numbers from A to B inclusively, in ascending order, if $A < B$, or in descending order, if $A \geq B$*
- ***c)*** ***Sum of N numbers:*** *N numbers are given in the input. Read them and print their sum.The first line of input contains the integer N, which is the number of integers to follow. Each of the next N lines contains one integer. Print the sum of these N integers.*
- ***d)*** ***Sum of Cubes:*** *For the given integer N calculate the following sum:*

$$1^3+2^3+\ldots+N^3$$

Lab Task 2

Factorial: *In mathematics, the factorial of an integer n, denoted by n! is the following product:*

$$n!=1\times2\times\ldots\times n$$

For the given integer n calculate the value n!

Lab Task 3

Number of zeros: *Given N numbers: the first number in the input is N, after that N integers are given. Count the number of zeros among the given integers and print it. You need to count the number of numbers that are equal to zero, not the number of zero digits.*

```
Input: 5 0 700 0 200 2          Output: 2
Input: 6 0 0 0 0 0 0            Output: 6
```

Lab Task 4

The length of Sequence: *Given a sequence of non-negative integers, where each number is written in a separate line. Determine the length of the sequence, where the sequence ends when the integer is equal to 0. Print the length of the sequence (not counting the integer 0). The numbers following the number 0 should be omitted.*

```
Input: 1 2 3 4 5 6 7 0 1 2 3        Output:7
```

Lab Task 5

The maximum of the Sequence: *A sequence consists of integer numbers and ends with the number 0. Determine the largest element of the sequence.*