UNIVERSITY OF THE CORDILLERAS
College of Information Technology and Computer Science
CC1 – Computing Fundamentals
Term & School Year: Third Trimester S.Y. 2022-2023
Faculty: Mailynn K. Sagayo

NAME:	Abenes, Enrico O.	**DATE:**	July 11, 2023
COURSE:	CC1 – INTL 1	**SCHEDULE:**	1:30 pm – 5:20 pm MT

Activity Number:

Activity Type: Laboratory Activity

TITLE: Loops

LEARNING OBJECTIVES:

At the end of this activity, the students should be able to:

1. Identify available loop control structures in Java.

2. Differentiate the different loop control structures in Java.

3. Implement these different loop control structures for a given problem.

4. Create a complete Java program that simulates these basic operations.

INSTRUCTIONS:

1. Make sure you have your own individual account/monitor.

2. Always save your work and log off when not using the computer.

3. By now you should have been familiarized with using your text editor/net beans.

4. By now you should know how to create, save, compile, execute, and debug programs in Java.

5. Use the skills and learning obtained in Midterms for you to successfully finish the learning objectives of this module.

DURATION: One to two Meetings

HANDS-ON:

1. Log on using your own individual account. Use your own username and

password.

2. Open your text editor/NetBeans.

3. Write your next Java program:

```
/* Programmed by: <write your name here>
   Program title: Sum1.java
   Program Date: <write the date today here> */

import java.io.*;
public class Sum1{
  public static void main(String[] args){
      int start = 0, end = 0, sum;
      String input = " ";

      BufferedReader in = new BufferedReader(new
                            InputStreamReader(System.in));

      try{
        System.out.print("Input starting number: ");
        input = in.readLine();
        start = Integer.parseInt(input);

        System.out.print("Input ending number: ");
        input = in.readLine();
        end = Integer.parseInt(input);
      }catch(IOException e){
        System.out.println("Error!");
      }

      if(start >= end){
        System.out.print("Starting number should be lesser ");
        System.out.println("than the ending number. ");
        System.out.println("Try again.");
        System.exit(0);
      }

      if(start%2 == 0)
      {
            start = start + 1;
      }

      sum = 0;
      while(start <= end)
      {
            sum = sum + start;
            start = start + 2;
      }
      System.out.println("Sum = " + sum);
  }
}
```

3.1. Write your next program by copying the source code shown below to your text editor.

3.2. Save the program as Sum1.java then compile your program until no errors and warnings are reported.

3.3. Run your program.

3.4. Simulate and write what will be displayed on the screen.

```
Input starting number: 3

Input ending number: 10

Sum = 24
```

4. Now let's try another implementation. Copy the source code below and save it as Sum2.java

```
/* Programmed by: <write your name here>
   Program title: Sum2.java
   Program Date: <write the date today here> */

import java.io.*;
public class Sum2{
  public static void main(String[] args){
      int start = 0, end = 0, sum;
      String input = " ";

      BufferedReader in = new BufferedReader(new
                          InputStreamReader(System.in));
      try{
        System.out.print("Input starting number: ");
        input = in.readLine();
        start = Integer.parseInt(input);

        System.out.print("Input ending number: ");
        input = in.readLine();
        end = Integer.parseInt(input);
      }catch(IOException e){
        System.out.println("Error!");
      }

      if(start >= end){
        System.out.print("Starting number should be lesser ");
        System.out.println("than the ending number. ");
        System.out.println("Try again.");
        System.exit(0);
      }
      if(start%2 == 0){
            start = start + 1;
      }
      sum = 0;
      for(start = start; start <= end; start = start + 2){
            sum = sum + start;
      }
      System.out.println("Sum = " + sum);
  }
}
```

UNIVERSITY OF THE CORDILLERAS
College of Information Technology and Computer Science
CC1 – Computing Fundamentals
Term & School Year: Third Trimester S.Y. 2022-2023
Faculty: Mailynn K. Sagayo

4.1. Simulate and write what will be displayed on the screen. Try using the same input values in your Sum1.java

```
Input starting number: 3

Input ending number: 10

Sum = 24
```

5. Let us try another implementation. Copy the source code and save it as Sum3.java

```
/* Programmed by: <write your name here>
   Program title: Sum3.java
   Program Date: <write the date today here> */

import java.io.*;
public class Sum3{
  public static void main(String[] args){
      int start = 0, end = 0, sum;
      String input = " ";

      BufferedReader in = new BufferedReader(new
                          InputStreamReader(System.in));
      try{
        System.out.print("Input starting number: ");
        input = in.readLine();
        start = Integer.parseInt(input);
        System.out.print("Input ending number: ");
        input = in.readLine();
        end = Integer.parseInt(input);
      }catch(IOException e){
        System.out.println("Error!");
      }
      if(start >= end){
        System.out.print("Starting number should be lesser ");
        System.out.println("than the ending number. ");
        System.out.println("Try again.");
        System.exit(0);
      }
      if(start%2 == 0){
            start = start + 1;
      }
      sum = 0;
      do{
            sum = sum + start;
            start = start + 2;
      }while(start <= end);

      System.out.println("Sum = " + sum);
  }
}
```

UNIVERSITY OF THE CORDILLERAS
College of Information Technology and Computer Science
CC1 – Computing Fundamentals
Term & School Year: Third Trimester S.Y. 2022-2023
Faculty: Mailynn K. Sagayo

5.1. Simulate and write what will be displayed on the screen. Try using the same input values in your Sum1.java and Sum2.java

```
Input starting number: 3

Input ending number: 10

Sum = 24
```

6. After your simulation of the three programs, what do you think is the main objective of these programs?

Calculating the total of all odd numbers in a certain range is the main goal of the three Java files. To do this, they gather user input for the range's starting and ending numbers and conduct checks to confirm the accuracy of the information. After that, they iterate across the range, adding each number to the total, using various looping constructions (while loop, for loop, and do-while loop). All three programs attempt to compute the sum of numbers inside the given range, although employing various looping algorithms.

7. Identify what are the different loop control structures in Java.

for loop – It enables you to use an initialization phrase, condition, and update statement with a code block to run it again for a predetermined number of times.

while loop – If a certain condition is true, a block of code is repeatedly executed.

do-while loop – The only difference between it and the while loop is that it checks the condition at the end to make sure the loop runs at least once.

8. What are the usual similar components of these loop constructs?

Condition – Before beginning each iteration, the condition is examined to determine whether the loop should be continued or ended.

Body – As long as the loop condition is true, this particular piece of code will be run endlessly.

Initialization – Before the loop starts, it requires initializing or setting up any necessary conditions or variables. It establishes the starting variables or conditions needed for the loop to begin.

Update – It specifies how to update or increase the loop control variable after each iteration. At the conclusion of each iteration, it describes the procedure to update the loop control variable.

Entry/Exit – Depending on the circumstance, it specifies how the loop is entered and exits.

9. What do you think is different among these identified loop constructs?

Justify your answer.

Java's **for, while, and do-while loops** are distinct from one another in terms of their structure, loop entry behavior, loop control, and usual usage.

The **for loop** employs a loop control variable, verifies the condition before each iteration, and enters the loop after setup.

The **while loop**'s structure is more adaptable since it checks the condition before going into the loop body. It is dependent on an outside circumstance and continues as long as it holds true.

Although it also has a flexible structure, the **do-while loop** performs the loop body first to guarantee that it is run at least once. After each iteration, the condition is checked.

10. Create a complete Java program that shall allow the user to accept three integer values representing the START, END, and STEP respectively. The START should always be lesser than the END value, and the STEP is always greater than zero. The program shall print vertically the values starting from the START to the last value which can be equal to or lesser than the END incremented by the STEP value.

Example Output 1: if START = 1, END = 10, STEP = 2

```
Input START value   = 1
Input END value     = 10
Input STEP value    = 2

1
3
5
7
9

Do you want to try again (Y/N)? Y
```

```
Input START value    = -10
Input END value      = 20
Input STEP value     = 5

-10
-5
0
5
10
15
20

Do you want to try again (Y/N)? N
```

Example Output 2: if START = -10, END = 20, STEP = 5

10.1. Your program should automatically detect any errors in the initial input.

10.2. Your program should have an additional feature that asks the user whether the user wants TO TRY AGAIN. The Program will not terminate until the user inputs any character other than 'Y' or 'y'.

10.3. Implement the program using all looping constructs identified earlier.

10.4. Write your complete programs in the space provided.

WHILE LOOP:

```
/* Programmed by: Abenes, Enrico O.
   Program Title: WhileLoop.java
   Program Date: July 11, 2023*/

package intl.cc1;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class WhileLoop {
    public static void main(String[] args) {
        int start, end, step;
```

UNIVERSITY OF THE CORDILLERAS
College of Information Technology and Computer Science
CC1 – Computing Fundamentals
Term & School Year: Third Trimester S.Y. 2022-2023
Faculty: Mailynn K. Sagayo

```
        while (tryAgain) {
            try {
                System.out.print("Input START value   = ");
                choice = basa.readLine();
                start = Integer.parseInt(choice);

                System.out.print("Input END value     = ");
                choice = basa.readLine();
                end = Integer.parseInt(choice);

                System.out.print("Input STEP value    = ");
                choice = basa.readLine();
                step = Integer.parseInt(choice);

                System.out.println();
                int i = start;
                while (i <= end) {
                    System.out.println(i);
                    i += step;
                }

                System.out.println();
                System.out.print("Do you want to try again (Y/N)? ");
                choice = basa.readLine().toUpperCase();
                tryAgain = choice.equals("Y");
            } catch (IOException | NumberFormatException e) {
                System.out.println("Error!");
                tryAgain = true;
            }
        }
    }
}
```

FOR LOOP:

```
/* Programmed by: Abenes, Enrico O.
   Program Title: ForLoop.java
   Program Date: July 11, 2023*/

package intl.cc1;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class ForLoop {
    public static void main(String[] args) {
        int start, end, step;
        String choice;
```

UNIVERSITY OF THE CORDILLERAS
College of Information Technology and Computer Science
CC1 – Computing Fundamentals
Term & School Year: Third Trimester S.Y. 2022-2023
Faculty: Mailynn K. Sagayo

```
        BufferedReader basa = new BufferedReader(new
InputStreamReader(System.in));

        for (choice = "Y"; choice.equals("Y"); ) {
            try  {
                System.out.print("Input START value   = ");
                choice = basa.readLine();
                start = Integer.parseInt(choice);

                System.out.print("Input END value     = ");
                choice = basa.readLine();
                end = Integer.parseInt(choice);

                System.out.print("Input STEP value    = ");
                choice = basa.readLine();
                step = Integer.parseInt(choice);

                System.out.println();
                for (int i = start; i <= end; i+= step) {
                    System.out.println(i);
                }

                System.out.println();
                System.out.print("Do you want to try again (Y/N)? ");
                choice = basa.readLine().toUpperCase();
            } catch (IOException | NumberFormatException e) {
                System.out.println("Error!");
                choice = "Y";
            }
        }
    }
}
```

DO-WHILE LOOP:

```
/* Programmed by: Abenes, Enrico O.
   Program Title: DoWhileLoop.java
   Program Date: July 11, 2023*/

package intl.cc1;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class DoWhileLoop {
    public static void main(String[] args) {
        int start, end, step;
        String choice;
```

UNIVERSITY OF THE CORDILLERAS
College of Information Technology and Computer Science
CC1 – Computing Fundamentals
Term & School Year: Third Trimester S.Y. 2022-2023
Faculty: Mailynn K. Sagayo

```
        BufferedReader basa = new BufferedReader(new
InputStreamReader(System.in));

        boolean tryAgain;
        do {
            try {
                System.out.print("Input START value   = ");
                choice = basa.readLine();
                start = Integer.parseInt(choice);

                System.out.print("Input END value     = ");
                choice = basa.readLine();
                end = Integer.parseInt(choice);

                System.out.print("Input STEP value    = ");
                choice = basa.readLine();
                step = Integer.parseInt(choice);

                System.out.println();
                int i = start;
                do {
                    System.out.println(i);
                    i += step;
                } while (i <= end);

                System.out.println();
                System.out.print("Do you want to try again (Y/N)? ");
                choice = basa.readLine().toUpperCase();
                tryAgain = choice.equals("Y");
            } catch (IOException | NumberFormatException e) {
                System.out.println("Error!");
                tryAgain = true;
            }
        } while (tryAgain);
    }
}
```

UNIVERSITY OF THE CORDILLERAS
College of Information Technology and Computer Science
CC1 – Computing Fundamentals
Term & School Year: Third Trimester S.Y. 2022-2023
Faculty: Mailynn K. Sagayo

Rubrics:

Code Content	The source code submitted covers all the items specified in the activity and satisfactorily meets all these requirements. It also makes use of the specific features of Java specified in the activity.	20 pts
Code Function	The source code submitted works completely with no errors and provides the correct expected output when run.	20 pts
Code Syntax	The source code submitted has sound logic and follows proper syntax for Java, with no unnecessarily complicated code.	10 pts
Total		**100 pts**