JOBSHEET 11

LOOPING 2

1. Objective

- Students are able to understand the concept of nested loop
- Students are able to explain nested loop writing format
- Students are able to implement nested loop flowchart using Java programming language

2. Laboratory

2.1 Experiment 1: Loop Review

1. This experiment is aimed at reviewing the loop that had been studied in the previous week. In experiment 1, a program will be made to make a view * N times sideways.
2. Create a new class with the name **Star** and save it in the **Star.java** file
3. Write the basic structure of the Java programming language which contains the **main()** function
4. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
5. In the **main()** function that has been created, declare a **Scanner** object with the name **sc**
6. On the next line, write the instructions for entering the value to be stored in variable **N**

```
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
```

7. On the next line, create a loop syntax using **for**

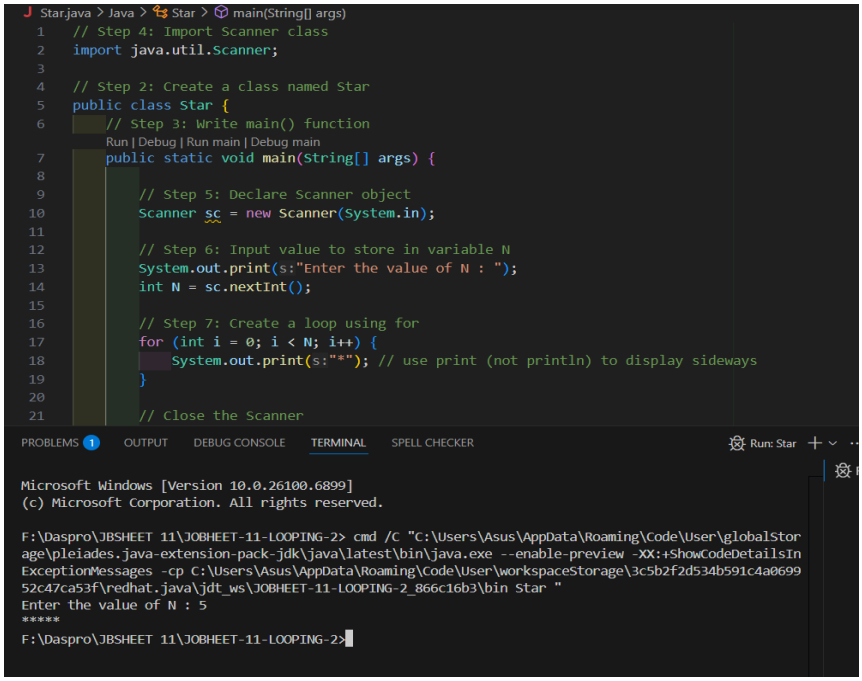
```
for (int i = 1; i <= N; i++) {
    System.out.print("*");
}
```

Note: *please note that the print command is used, not println because we want to display without any new lines*

8. Compile and run the program.

9. Observe the results, match the results of the running programs that you have created according to the following display

```
Enter the value of N: 5
*****
```



Questions!

1. If in **for** loop, the initialization **i = 1** is changed to **i = 0**, what is the result? How can It be like that?

If we change i=1 is changed to i=0 so its prints start instead of 5,

```
for (int i = 0; i < N; i++) {
  System.out.print("*");
  }
```

2. If in **for** loop, condition **i <= N** is changed to **i > N**, what is the result? How can It be like that?

   If you reverse the direction of counting(e.g., start from a large number and decrease i), then i > N can make sense.This would print *****, because i starts from 5 and decreases until it's no longer.

3 . If in **for** loop, the condition for step **i++** is changed to **i--** what is the result? How can

It be like that?

The condition never becomes false because i keeps decreasing forever (1, 0, -1, -2, -3, …) and it will always be ≤ N (5)

2.2 Experiment 2: Square Star

1. In Experiment 2, an experiment was carried out on nested loops. The case that will be solved is to create a square display *, with side lengths of N. Suppose N is entered as 5, then the results are as follows:

   ```
   *****
   *****
   *****
   *****
   *****
   ```

2. If you look closely, it's actually like the case of Experiment 1, isn't it? In Experiment 1, for example, the input N has a value of 5, then what will be produced is ***** (we can think of this as an **inner loop** that displays 5 stars *****), then for the case of Experiment 2, this is not the result of experiment 1 does it just need to be repeated N times? (by adding an **outer loop** to repeat the **inner loop** process N times)
3. Create a new class with the name **Square** and save it in the **Square.java** file
4. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
5. Create a **main()** function, and add the same program code as the contents of the **main()** function in Experiment 1

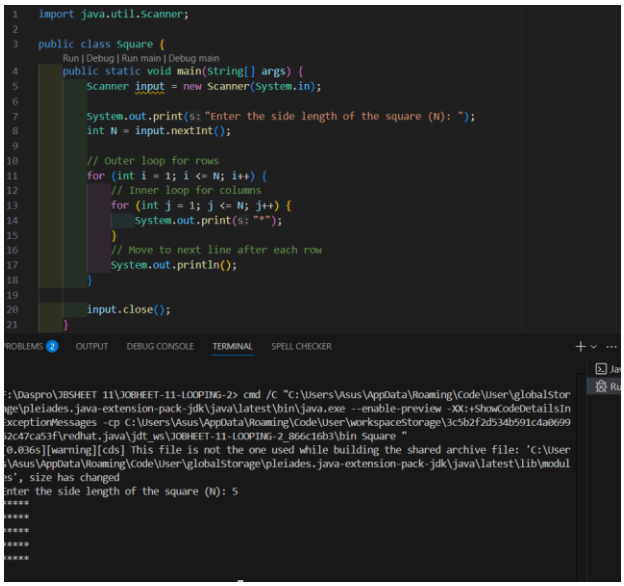
```
Scanner sc = new Scanner(System.in);
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
for (int i = 1; i <= N; i++) {
    System.out.print("*");
}
```

6. Compile and run the program. Make sure the results given are the same as in Experiment 1
7. Pay attention to the iterative syntax used to print * N times sideways. In step 5, we make **for** loop structure (red box) as an **inner loop**
8. Furthermore, the inner loop needs to be repeated N times to display the * symbol to generate output like step 1. Thus, it is necessary to add an **outer loop**

```
for (int iOuter = 1; iOuter <= N; iOuter++) {
    for (int i = 1; i <= N; i++) {
        System.out.print("*");
    }
    System.out.println("");
}
```

9. Compile and run the program.
10. Observe the results, match the results of the running programs that you have created according to the following display

```
Enter the value of N: 5
*****
*****
*****
*****
*****
```



Questions!

1. Pay attention to outer loop. If in **for** syntax, the initialization **iOuter = 1** is changed to **iOuter = 0**, what is the result? How can it be like that?

   The result iOuter 1 correct square size,iOuter 0 Extra row printed
   If N = 5, then iOuter will takes the values : (0, 1, 2, 3, 4, 5,) → 6 total times
   So instead of printing 5 rows, the program will 6 print rows of *.

2. Return the program to normal with initialization **iOuter = 1**. Then pay attention to the inner loop. If in **for** syntax, the initialization **i = 1** is changed to **i = 0**, what is the result? How can it be like that?

   The square is 6x5 instead of 5x5. So It gives the correct size even when starting form 0 – a useful trick to know

3. What is the difference between outer loop and inner loop?

   - The outer loop
   - Controls how many rows are printed.
   - Runs once for each line of output.
   - Every time the outer loop runs once, the inner loop runs completely.

   - Example:
   If N = 5, the outer loop runs 5 times — meaning 5 rows will be printed

   -The inner loop
   - Controls how many columns (symbols *) are printed within each row.
   - Runs multiple times for each single iteration of the outer loop.
   - After the inner loop finishes, the outer loop moves to the next row.

   Example:
   If N = 5, the inner loop prints 5 stars per row, before the program goes to the next line.

4. Why is it necessary to add the syntax **System.out.println();** under inner loop? What will happen if the syntax is omitted?

   Because it moves the cursor to the next line after one full row of stars is printed. Without it, all stars would print on the same line. Its helps visualize why the line break is so important.

5. Commit and push the changes to GitHub

2.3 Experiment 3: Triangle Star

1. In Experiment 3, a right-angled triangle with a height of N was carried out. Suppose N is entered as 5, then the results are as follows:

```
*
**
***
****
*****
```

2. Create a new class with the name **Triangle** and save it in the **Triangle.java** file
3. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
4. Create a **main()** function, and fill in the following program code into the **main()** function

```
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
int i = 0;
while (i <= N) {
    int j = 0;
    while (j < i) {
        System.out.print("*");
        j++;
    }
    i++;
}
```

5. Compile and run the program. Observe the results!

```java
import java.util.Scanner;

public class Triangle {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        // 4. Create Scanner object to take input from user
        Scanner input = new Scanner(System.in);

        System.out.print(s: "Enter the height of the triangle (N): ");
        int N = input.nextInt(); // read integer input

        // Loop to print right-angled triangle
        for (int i = 1; i <= N; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(s: "*");
            }
            System.out.println(); // move to next line
        }

        input.close();
    }
}
```

```
ROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   SPELL CHECKER

icrosoft Windows [Version 10.0.26100.6899]
c) Microsoft Corporation. All rights reserved.

:\Daspro\JBSHEET 11\JOBHEET-11-LOOPING-2> cmd /C "C:\Users\Asus\AppData\Roaming\Code\User\globalStor
ge\pleiades.java-extension-pack-jdk\java\latest\bin\java.exe --enable-preview -XX:+ShowCodeDetailsIn
xceptionMessages -cp C:\Users\Asus\AppData\Roaming\Code\User\workspaceStorage\3c5b2f2d534b591c4a0699
2c47ca53f\redhat.java\jdt_ws\JOBHEET-11-LOOPING-2_866c16b3\bin Triangle "
0.022s][warning][cds] This file is not the one used while building the shared archive file: 'C:\User
\Asus\AppData\Roaming\Code\User\globalStorage\pleiades.java-extension-pack-jdk\java\latest\lib\modul
s', size has changed
nter the height of the triangle (N): 5

*
**
***
****
```

Questions!

1. Look at the results, does the output produced with a value of N = 5 match the following display?

```
*
**
***
****
*****
```

Yess output is correct

2. If not, which parts should be improved or added? Describe any parts that need to be improved or added!

   No,I think its correct

2.4 Experiment 4: Guess the Number Quiz

1. Create a new class with the name **Quiz** and save it in the **Quiz.java** file
2. Add Scanner and Random libraries in the top line of the program

```
import java.util.Scanner;
import java.util.Random;
```

3. Create a **main()** function
4. In the **main()** function, make a **Scanner** declaration with the name **input** and **Random** declaration with the name **rand**. In this case, Random is used to randomize the numbers

```
Scanner input = new Scanner(System.in);
Random rand = new Random();
```

5. Then on the next line, add the following syntax

```
char menu = 'y';
do {
    int number = rand.nextInt(10) + 1;
    boolean success = false;
    do {
        System.out.print("Guess the number (1-10): ");
        int answer = input.nextInt();
        input.nextLine();
        success = (answer == number);
    } while (!success);
    System.out.print("Do you want to repeat the game (Y/N)");
    menu = input.next().charAt(0);
    input.nextLine();
} while (menu == 'Y' || menu == 'y');
```

Note: the **input.nextLine()** syntax in that snippet is used to ignore the new line

character

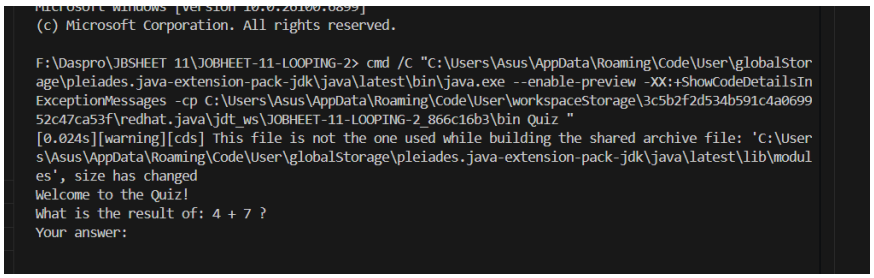
6. Compile and run the program. Observe the results!

Questions!

1. Explain the program flow in Experiment 4!

   Step 1 Program start, steps 2 Importing Libraries, steps 3 Creating object, steps 4 Generating Number, steps 5 Displaying the Quiz Question, steps 6 getting user input, steps 7 checking the answer , last steps program ends

2. What must be done to discontinue (not repeat) the game?
   - **Add a loop** (for example, a while or do-while loop) that keeps the game running.
   - **Ask the user** after each round whether they want to play again.
   - **Break** the loop if the user answers something like "No".
3. Modify the program above, so that it can display information about: input the guess value entered by the user, whether it is smaller or greater than the answer (number) randomly determined by the computer!



4. Commit and push the changes to GitHub

2.5 Experiment 5: Filling and Displaying Arrays

1. Create a new class with the name **NestedLoopStudentID**
2. Create a **main()** function
3. In the **main()** function, add a declaration for Scanner named **scanner** and a 2-dimensional array declaration named **temps** with 5 rows and 7 columns of type **double**
4. Add the following program code

```
for (int i = 0; i < temps.length; i++) {
    System.out.println("City: " + i);
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print("Day " + (j + 1) + ": ");
        temps[i][j] = scanner.nextDouble();
    }
    System.out.println();
}
```

5. Then, also add the following several lines of program code

```
for (int i = 0; i < temps.length; i++) {
    System.out.println("City: " + i);
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print(temps[i][j] + " ");
    }
    System.out.println();
}
```

6. Compile and run the program. Observe the results!

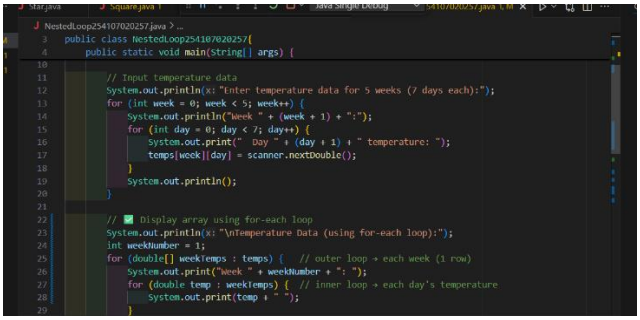
   Note: *This program example is a case example for storing temperature data in a city. Check back to last week's material slides.*

```
Temperature Data:
Week 1: 30.5 31.0 32.2 33.1 31.5 30.0 29.8
Week 2: 28.5 29.0 30.2 31.0 32.1 30.8 29.7
Week 3: 33.0 32.8 34.1 35.0 34.5 33.2 32.0
Week 4: 27.5 28.1 29.3 30.0 31.2 30.8 29.9
Week 5: 26.0 27.0 28.5 29.2 28.7 27.8 26.9

Average Temperature per Week:
Week 1: 31.157142857142862
Week 2: 30.185714285714287
Week 3: 33.51428571428572
Week 4: 29.542857142857144
Week 5: 27.72857142857143
```

Questions!

1. Explain the program flow in Experiment 5!
   - Enables keyboard input.
   - Prepares tools to store temperature data.
   - Users enters temperature for each day(Nested Loops).
   - Prints all recorded temperatures.
   - Computes average temperature for each week
   - Shows weekly averages.
   - Ends program safely
2. Modify the program to display an array using foreach!

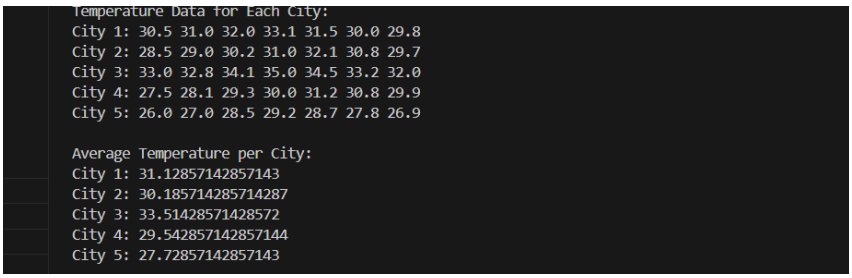


```
public class NestedLoop25410702025{
    public static void main(String[] args) {

        // Input temperature data
        System.out.println("Enter temperature data for 5 weeks (7 days each):");
        for (int week = 0; week < 5; week++) {
            System.out.println("Week " + (week + 1) + ":");
            for (int day = 0; day < 7; day++) {
                System.out.print("  Day " + (day + 1) + " temperature: ");
                temps[week][day] = scanner.nextDouble();
            }
            System.out.println();
        }

        // ✅ Display array using for-each loop
        System.out.println("\nTemperature Data (using for-each loop):");
        int weekNumber = 1;
        for (double[] weekTemps : temps) {  // outer loop → each week (1 row)
            System.out.print("Week " + weekNumber + ": ");
            for (double temp : weekTemps) {  // inner loop → each day's temperature
                System.out.print(temp + " ");
            }
```

3. Modify the program so that it can display the average value for each city!



```
Temperature Data for Each City:
City 1: 30.5 31.0 32.0 33.1 31.5 30.0 29.8
City 2: 28.5 29.0 30.2 31.0 32.1 30.8 29.7
City 3: 33.0 32.8 34.1 35.0 34.5 33.2 32.0
City 4: 27.5 28.1 29.3 30.0 31.2 30.8 29.9
City 5: 26.0 27.0 28.5 29.2 28.7 27.8 26.9

Average Temperature per City:
City 1: 31.12857142857143
City 2: 30.185714285714287
City 3: 33.51428571428572
City 4: 29.542857142857144
City 5: 27.72857142857143
```

4. Commit and push the changes to GitHub

3. Assignment

1. Create a program to print a numeric triangle display as below based on the N input (minimum N value is 3). Example N = 5

```java
import java.util.Scanner;

public class NumericTriangle {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print(s: "Enter N (minimum 3): ");
        int N = input.nextInt();

        if (N < 3) {
            System.out.println(x: "N must be at least 3.");
            return;
        }

        for (int i = 1; i <= N; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(j);
            }
            System.out.println();
        }
    }
}
```

```
    1
   12
  123
 1234
12345
```

2. Create a program to print the star triangle view shown below based on the N input (minimum N value is 5). Example N = 7

```
(C) Microsoft Corporation. All rights reserved.

F:\Daspro\JBSHEET 11\JOBHEET-11-LOOPING-2> cmd /C "C:\Users\Asus\AppData\Roaming\Code\User\globalStor
age\pleiades.java-extension-pack-jdk\java\latest\bin\java.exe --enable-preview -XX:+ShowCodeDetailsIn
ExceptionMessages -cp C:\Users\Asus\AppData\Roaming\Code\User\workspaceStorage\3c5b2f2d534b591c4a0699
52c47ca53f\redhat.java\jdt_ws\JOBHEET-11-LOOPING-2_866c16b3\bin StarTriangle "
[0.027s][warning][cds] This file is not the one used while building the shared archive file: 'C:\User
s\Asus\AppData\Roaming\Code\User\globalStorage\pleiades.java-extension-pack-jdk\java\latest\lib\modul
es', size has changed
Enter N (minimum 5): 7
*
**
***
****
*****
******
```

```
*******
******
*****
****
***
**
*
```

3. Create a program to print a square numeric display like the one below based on N input (minimum N value is 3). Example N = 3 and N = 5

```
            5 5 5 5 5
            5       5
3 3 3       5       5
3   3       5       5
3 3 3       5 5 5 5 5
```

```
F:\Daspro\JBSHEET 11\JOBHEET-11-LOOPING-2> cmd /C "C:\Users\Asus\AppData\Roaming\Code\User\globalStor
age\pleiades.java-extension-pack-jdk\java\latest\bin\java.exe --enable-preview -XX:+ShowCodeDetailsIn
ExceptionMessages -cp C:\Users\Asus\AppData\Roaming\Code\User\workspaceStorage\3c5b2f2d534b591c4a0699
52c47ca53f\redhat.java\jdt_ws\JOBHEET-11-LOOPING-2_866c16b3\bin SquareNumeric "
[0.019s][warning][cds] This file is not the one used while building the shared archive file: 'C:\User
s\Asus\AppData\Roaming\Code\User\globalStorage\pleiades.java-extension-pack-jdk\java\latest\lib\modul
es', size has changed
Enter N (minimum 3): 3
3 3 3
3 3 3
3 3 3
```

```
d /C "C:\Users\Asus\AppData\Roaming\Code\User\globalStorage\pleiades.java-extension-pack-jdk\java\lat
est\bin\java.exe --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Asus\AppData\R
oaming\Code\User\workspaceStorage\3c5b2f2d534b591c4a069952c47ca53f\redhat.java\jdt_ws\JOBHEET-11-LOOP
ING-2_866c16b3\bin SquareNumeric "
[0.022s][warning][cds] This file is not the one used while building the shared archive file: 'C:\User
s\Asus\AppData\Roaming\Code\User\globalStorage\pleiades.java-extension-pack-jdk\java\latest\lib\modul
es', size has changed
Enter N (minimum 3): 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
```

4. In 2024, Malang State Polytechnic will host the Porseni national event. There are several sports that are competed in, such as **badminton**, **table tennis**, **basketball**, and **volleyball**. Each sport sends its **5 best athletes** from all polytechnics throughout Indonesia to take part in this biannual event. Create a data storage to display

information on the **names of athletes** from the various branches mentioned in **ascending** order.

```java
import java.util.Arrays;

public class PorseniAthletes {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        String[] badminton = {"Andi", "Budi", "Citra", "Dewi", "Eka"};
        String[] tableTennis = {"Fajar", "Gilang", "Hana", "Ira", "Joko"};
        String[] basketball = {"Kevin", "Lina", "Mila", "Nando", "Oki"};
        String[] volleyball = {"Putra", "Qori", "Rina", "Sandi", "Tono"};

        String[][] allSports = {badminton, tableTennis, basketball, volleyball};
        String[] sportNames = {"Badminton", "Table Tennis", "Basketball", "Volleyball"};

        for (int i = 0; i < allSports.length; i++) {
            Arrays.sort(allSports[i]);
            System.out.println("\n" + sportNames[i] + " Athletes (sorted):");
            for (String name : allSports[i]) {
                System.out.println(" - " + name);
            }
        }
    }
}
```

5. Implement the flowchart of the features you created in the previous theory assignment about nested loops!

```java
import java.util.Scanner;

public class NestedLoops {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n=== Nested Loop Pattern Menu ===");
            System.out.println("1. Numeric Triangle");
            System.out.println("2. Star Triangle");
            System.out.println("3. Square Number");
            System.out.println("4. Exit");
            System.out.print("Choose: ");
            choice = input.nextInt();

            switch (choice) {
                case 1 -> {
                    System.out.print("Enter N: ");
                    int N = input.nextInt();
                    for (int i = 1; i <= N; i++) {
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL SPELL CHECKER 15

6. Don't forget, all program code must be pushed to your repository.