JOBSHEET 7
LOOP 1

Name:Dava Rizky H
NIM:254107020246

1. Learning Outcome

After finishing this topic, students must be able to:

- explain the format of loop programming part 1
- implement a loop part 1 flowchart using the Java programming language

2. Labs Activity

2.1 Lab 1: Counting Multiples Using FOR

Time: 60 minutes

In this experiment, the code is created to display multiples of a specific number within the range of 1 to 50 using a FOR loop, and to calculate the total of these numbers.

1. Open text editor. Create a new Java File named **ForMultiplesNoAbsen.java**
2. Create the basic structure of Java program containing **class** declaration and **main()** method
3. Add the **Scanner** library.
4. Create or declare variable named **input** from **Scanner** library.
5. Create **int** variables named **multiple**, **sum**, and **counter**. Initialize variable **sum** and **counter** with 0.
6. Add the following code to get the user input!

```
System.out.print(s:"Input the multiple: ");
multiple = input.nextInt();
```

7. Create the FOR loop with IF condition to evaluate the multiples number

```
for(int i=1; i <= 50; i++) {
    if(i % multiple == 0) {
        sum = sum + i;
        counter++;

    }
}
```

8. Display the sum and counter of multiples number in range from 1 to 50.

```
System.out.printf(format:"There are %d number that are multiple of %d in range 1 to 50.\n", counter, multiple);
System.out.printf(format:"The sum of all multiples of %d in range 1 to 50 is %d. \n", multiple, sum);
```

9. Run the program and analyze the result. Your result must be like this:

```
Input the multiple: 5
There are 10 number that are multiple of 5 in range 1 to 50.
The sum of all multiples of 5 in range 1 to 50 is 275.
```

Questions

1. There are 3 main components in FOR loop. Based on experiment 1 above, identify and explain these 3 components!
2. Explain how the following code works!

```
for(int i=1; i <= 50; i++) {
    if(i % multiple == 0) {
        sum = sum + i;
        counter++;

    }
}
```

3. Modify the existing code by adding a new variable to calculate the average of all the specified multiples!
4. Create a new Java program file named **WhileMultiplesNoAbsen.java**.

2.2 Lab 2: Show Multiplication of 2

Times: 40 minutes

We want to create a program which take any number more than 2 and print the multiplication of 2 within given input.

1. Open the text editor and create a new Java file named **DisplayTwoNoAbsen.java**
2. Create the basic structure of Java program containing **class** declaration and **main**() method
3. Add the **Scanner** library.
4. Create or declare variable named **input** from **Scanner** library.
5. Declare **int** variable named **numInput**.
6. Add the following code to input from user.

```
System.out.print(s:"Input some number: ");
numInput = input.nextInt();
```

7. Add this following FOR loop.

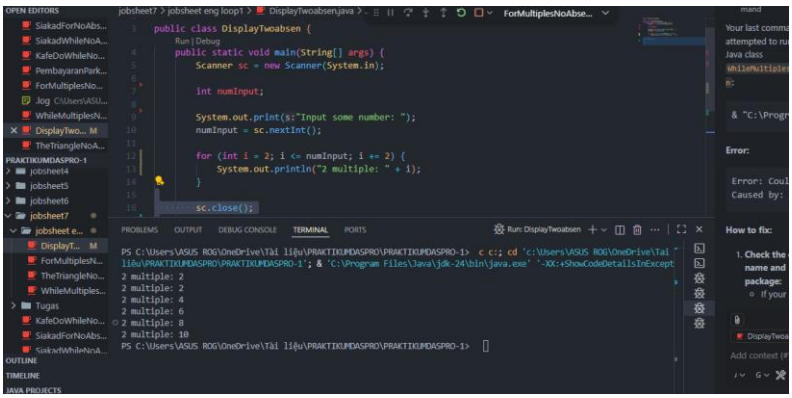
```
for(int i = 1; i <= numInput; i++) {
    if(i % 2 == 0) {
        System.out.println("2 multiple: "+i);
    }
}
```

8. Compile and run the program.
9. Expected result:

```
Input some number: 10
2 multiple: 2
2 multiple: 4
2 multiple: 6
2 multiple: 8
2 multiple: 10
```

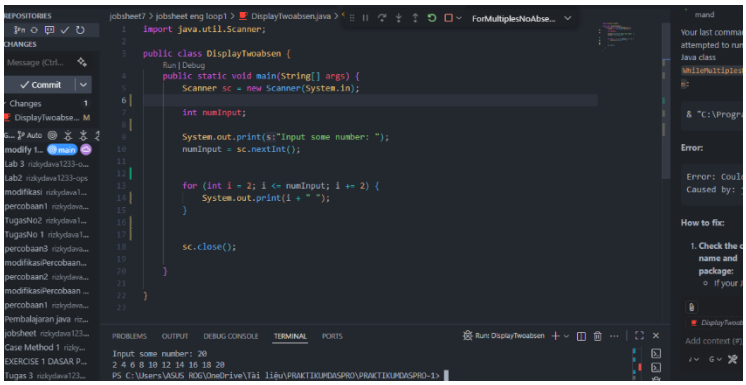
Questions

1. Do modification to make the program produce similar result but **WITHOUT IF** statement. Please insert a screenshot of your code to the report.



2. Do modification to make the program print like this following result. Please insert a screenshot of your code to the report.

```
Input some number: 10
2 4 6 8 10
```



2.3 Lab 3: The Triangle

Duration: 30 minutes

In this lab, you will create a rectangle pattern using single loop.

1. Open the text editor and create a new Java file called **TheTriangleNoAbsen.java**.
2. Create the basic structure of Java program containing class declaration and main method.
3. Add the **Scanner** library.
4. Create a variable **input** from Scanner library.
5. Create int variable called **numInput**.
6. Create int variable called **i** and give 0 as initial value.
7. Create String variable called **s** and give **empty string** as initial value.
8. Add this following code to take the input from user.

```
System.out.print(s:"Input some number: ");
numInput = input.nextInt();
```

9. Add this following loop, to print the rectangle pattern.

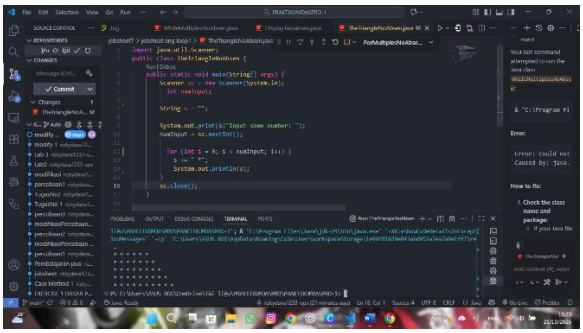
```
while (i < numInput) {
    s += " *";
    System.out.println(s);
    i++;
}
```

10. The expected result:

```
Input some number: 5
 *
 * *
 * * *
 * * * *
 * * * * *
```

Question

1. Do a modification on the program therefore your program utilize FOR statement rather than WHILE statement.



2. Explain the meaning of `s += " *"` and why is it possible?

   s += " *" means "add a space and an asterisk to the end of s".

   It works because Java supports string concatenation using +.

2.4. Lab 4: Calculating Leave Entitlement Using DO-WHILE

Times: 50 minutes

In this experiment, a program code is created using DO-WHILE to calculate the **leave entitlement** of an employee. Employees are entitled to 5 days of leave. Leave days will be deducted each time they are used. When there are only 2 days of leave remaining, the employee receives a warning to stop using their leave

1. Open the text editor and create a new Java file named **DoWhileLeaveEntitlementNoAbsen.java**
2. Create the basic structure of Java program containing class declaration and main() method
3. Add the **Scanner** library.
4. Create or declare variable named **input** from **Scanner** library.
5. Create variables **leaveEntitlement** and **numLeave** with int datatype.
6. Create variable **confirmation** with **String** datatype.
7. Create a DO-WHILE loop structure to get the user input from the keyboard and calculate leave entitlement

```
do {
    System.out.print(s:"Do you want to take a leave (y/n)? ");
    confirmation = input.next();

    if(confirmation.equalsIgnoreCase(anotherString:"y")) {
        System.out.print(s:"How many day(s)? ");
        numLeave = input.nextInt();

        if(numLeave <= leaveEntitlement) {
            leaveEntitlement -= numLeave;
            System.out.println("Remaining leave entitlement: "+leaveEntitlement);
        } else {
            System.out.println(x:"You dont have enough leave entitlement");
            break;
        }
    }
} while(leaveEntitlement > 0);
```

8. Run the program and analyze the result. It must be the same with the following output.

```
Input The Number of Leave Entitlement: 10
Do you want to take a leave (y/n)? y
How many day(s)? 7
Remaining leave entitlement: 3
Do you want to take a leave (y/n)? y
How many day(s)? 5
You dont have enough leave entitlement
```

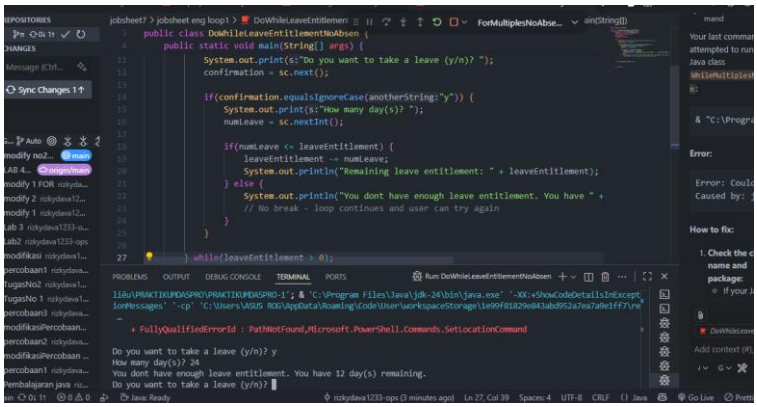
Questions

1. What is the use of the BREAK within the loop syntax?

   The BREAK function in a loop

   Break is used to forcibly terminate a loop before the loop condition completes. In this code, break terminates the program when the user requests more leave than is available.

2. Modify the program so that if the number of leave days requested is greater than the remaining entitlement, the program does not stop, allowing the user to enter the number of days according to the entitlement.



3. Commit and push the program code to GitHub.
4. When typing "t" as the confirmation input, what happens? Why?

   When the user types "t" as confirmation:

   * The program will skip the

if(confirmation.equalsIgnoreCase("y")) block because "t" ≠ "y"

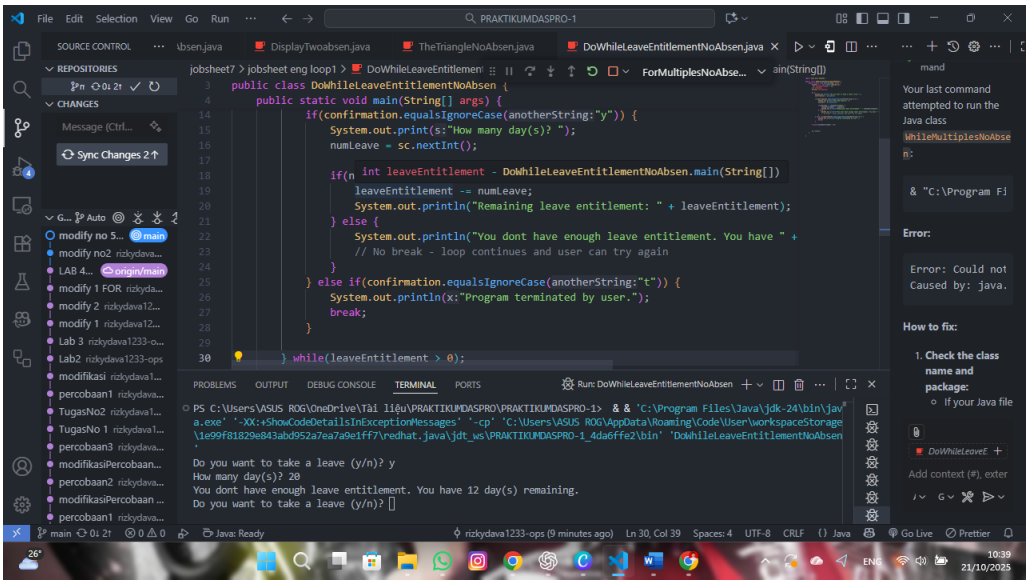
* The loop will continue because the condition

while(leaveEntitlement > 0) is still true

* The program will continue asking "Do you want to take a leave (y/n)?" endlessly (infinite loop)

Reason: There is no condition to stop the loop other than the leave expires or break.

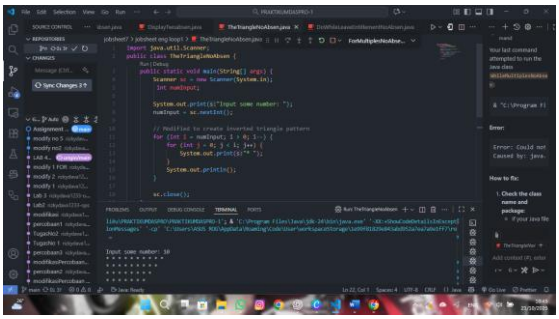
5. Modify the program code so that when the user enters "t" as the confirmation input, the program will stop.



Assignment (Open Challenge)

1. Do a modification on the program from the lab 3, so you get this following result,

```
Input some number: 5
* * * * *
* * * *
* * *
* *
*
```




POLITEKNIK NEGERI MALANG