**Practicum Report**

Job sheet 5

LOOPING 1

Experiment



MUHAMMAD BAIHAQI AULIA ASY’ARI

2241720145

CLASS 1I (INTERNATIONAL)

INFORMATICS ENGINEERING­­

INFORMATICS TECHNOLOGY

STATE POLYTECHNIC OF MALANG

Contents

[Laboratory Experiment 3](#_Toc116565639)

[Experiment 1 3](#_Toc116565640)

[Questions! 4](#_Toc116565641)

[Answer! 4](#_Toc116565642)

[Experiment 2 5](#_Toc116565643)

[Question! 7](#_Toc116565644)

[Answer! 7](#_Toc116565645)

# Laboratory Experiment

## Experiment 1

1. Open a text editor. Create a new file, name it factorialFor.java
2. Write the basic structure of the Java programming language which contains the main() function
3. Add the Scanner library.
4. Make a Scanner declaration with the name input
5. Create multiple int type variables with names number, factorial, and i
6. Write down the syntax for entering the value from keyboard

### Questions!

1. Describe the following syntax functions!
2. Modify the program code in Experiment 1 so that if the entered value is less than 0 the output "Value you entered is less than 0" and if the entered value is more than 100 the output will display "The value you entered is more than 100"!
3. Change the **&&** operator to **||** on **if (value >= 0 && value <= 100)**. Compile and run the program by entering the value = 105 using keyboard. Watch what happened! Why is the result like that?

### Answer!

1. It functions to filter number that are between the number 0 to 100 by including number greater-than or equal to 0 and number less-than or equal to 100
2. -
3. -Because the || operator makes any number bigger than 0 or less than 100, which is essentially every number that exist, pass through the if selection condition. Thus, any number can pass trough and further filtered through the nested selection and any number less 0 or more than 100 doesn’t have designated filter, it can only pass through the last else statement. And because the first if statement always will be passed by any number, the continuing if statement doesn’t even have the chance to be used, essentially rendering the last few lines redundant.

## Experiment 2

1. Observe the following flowchart! The flowchart is used to calculate a person's net salary after taxes according to their category (worker and businessman) and the amount of income.
2. Open a text editor. Create a new file, name it **Nested2.java**
3. Write the basic structure of the Java programming language which contains the **main()** function
4. Add the Scanner library.
5. Make a **Scanner** declaration with the name **sc**
6. Declare **category**, **income**, **netSalary**, and **tax** variables
7. Write down the syntax for entering the value from keyboard
8. Create a nested selection structure. The first check is used to check the category (worker or businessman). Then a second check is carried out to determine the amount of tax based on the income that has been entered. Then add the program code to calculate the net salary received after taxes
9. Compile and run the program. Observe the results!

### Question!

1. Run the program by entering category = worker and income = 2048485 using keyboard. Watch what happened! Why is the decimal number not displayed?
2. Describe the function of (int) in the following syntax!
3. Run the program by entering category = BUSINESSMAN and income = 2000000. Watch what happens! What are the uses of **equalsIgnoreCase**?
4. Change **equalsIgnoreCase** to **equals**, then run the program by entering category = BUSINESSMAN and income = 2000000. Watch what happens! Why is the result like that? What are the uses of **equals**?

### Answer!

1. The output is in int, because the netSalary is in int.
2. It narrows down the result of the (income – (income \* tax)) to integer.
3. -
4. -