

Nama : Azaria Cindy Sahasika

Nim : 2341760169 / 07

Kelas : 1G SIB

Lesson : Dasar Pemrograman

Material : Jobsheet 6

# JOBSHEET 6

**Selection part 2**

1. **Objective**
   * Students must have a good understanding on logical operators
   * Students must be able to solve logical problems using nested selection syntax
   * Students must able to create a Java program that utilizes nested selection syntax

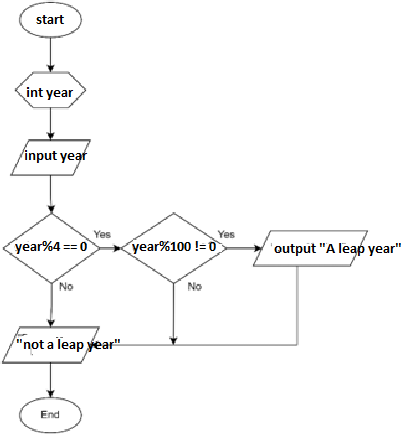
# Laboratory

* 1. **Experiment 1**

**Experiment Time: 50 minutes**

1. Determining a leap year. The basic rule for a leap year is a multiple of 4 and not a multiple of 100.

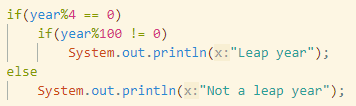
The following flowchart is an algorithm for determining leap years



1. Open your project folder in VSCode, and create a new Java file named

## Selection2Exp1StudentIDNumber.java

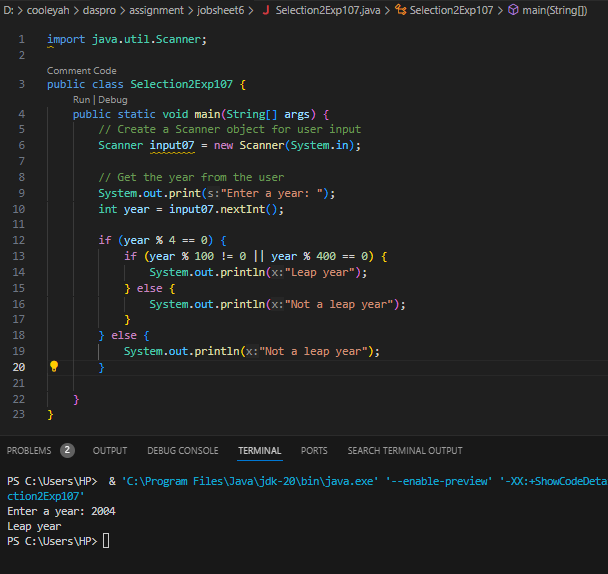
1. Create a basic structure for Java program (including class and method main)
2. Add import statement fo Scanner library
3. Create a new object from Scanner named **inputStudentIDNumber**
4. Add a statement to get the user input.
5. Create a selection structure as follows:



1. Run the program and observe the result:



1. Commit and push the changes into your repository.



# Question

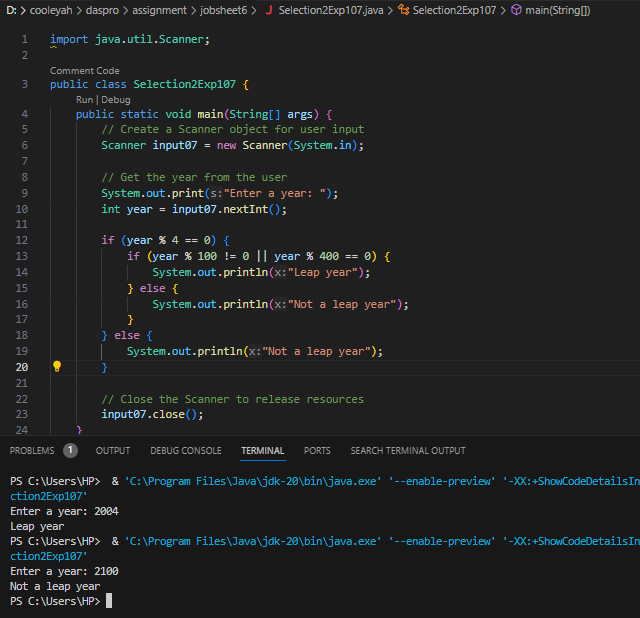
1. What is the output if the input is 2100? Please explain your answer! How to ensure that output complies with regulations (2100 is not a leap year)?
2. The rule for determining leap years is as follows:

* If a year is divisible by 4, then it is a leap year.
* However, if the year is also divisible by 100, then the year is not a leap year
* Unless, if the year is also divisible by 400, then it is a leap year.

1. In the case of 2100:

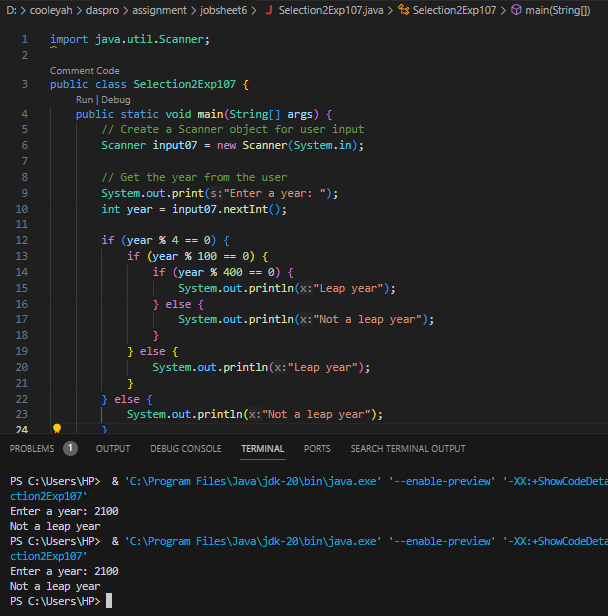
* Divisible by 4 because 2100 % 4 equals 0.
* Divisible by 100 because 2100 % 100 equals 0.
* Not divisible by 400 because 2100 % 400 does not equal 0.

1. Modify the program according to answer number 1!



## Commit and push the changes into your repository!

1. The year 2000 is a multiple of 4 and a multiple of 100, but it is a leap year. So that, there is an additional rule to determine leap year. If the year is a multiple of 100 and is also a multiple of 400 then that year is a leap year. Modify the program to adjust to these rule! (Create the algorithm without using logical operators)

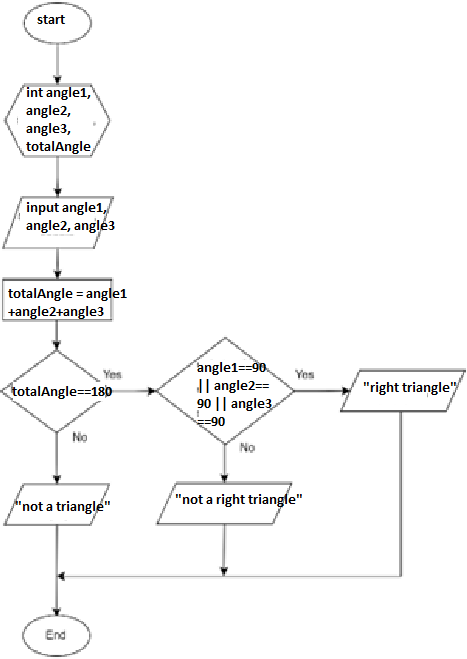


1. Commit and push the modifications to the repository!

# Experiment 2

**Experiment Time: 50 minutes**

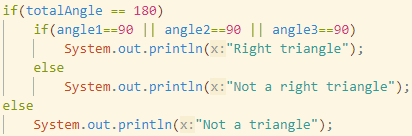
1. Determining the type of triangle, based on the 3 angles input.
2. Look at the following flowchart!



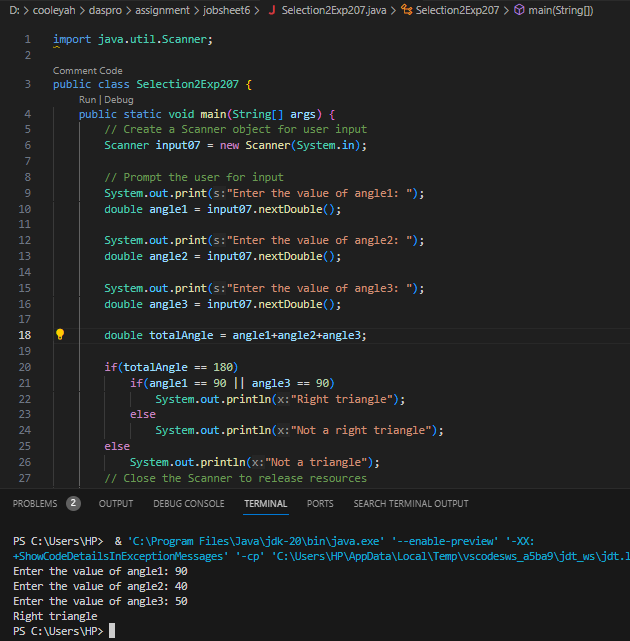
1. Create a new Java file named **Selection2Exp2StudentIDNumber.java**
2. Create a basic structure for java program, including class declaration and main method declaration
3. Import Scanner library.
4. Declare a new object for Scanner named **inputAbsen**
5. Write code to get user input for **angle1**, **angle2** and **angle3**
6. Add the following statement to calculate the **totalAngle**



1. Create a selection structure as follows :

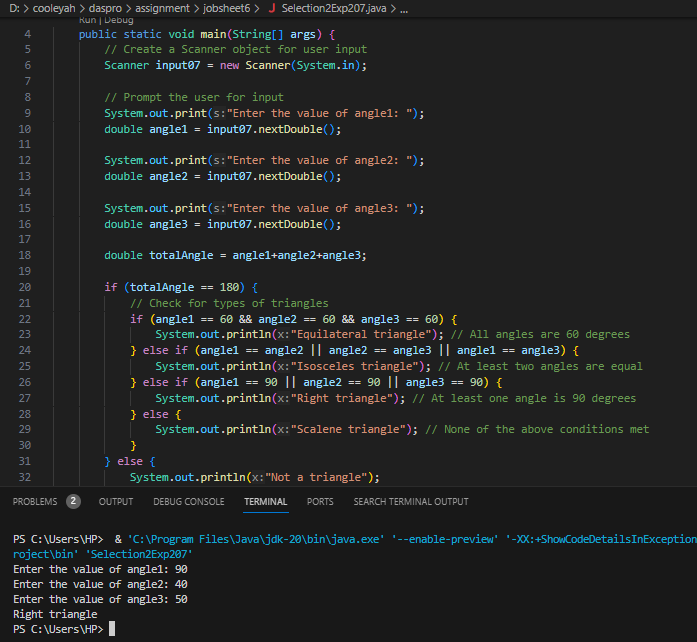


1. Run the program and observer the output:
2. Commit and push the changes into your repository.



# Question

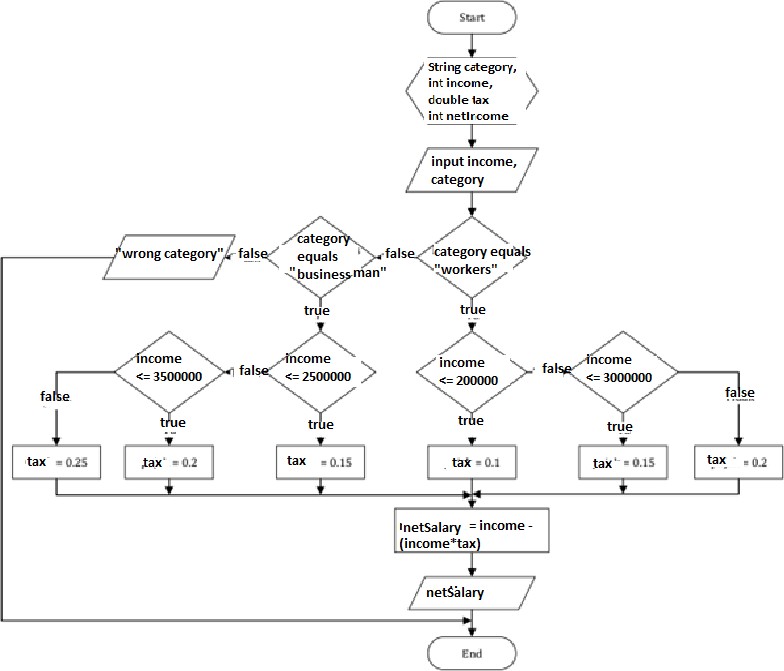
1. Modify the source code, so that it can detect the other types of triangles (equilateral triangle and isosceles triangle)



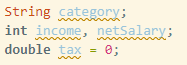
# Experiment 3

**Experiment Time: 40 minutes**

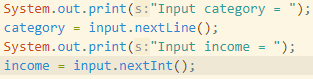
1. Look at the flowchart below, this flowchart is used to calculate a person's **net salary** after **tax** deductions according to their **category** (**workers** and **businessman**) and the amount of **income**.



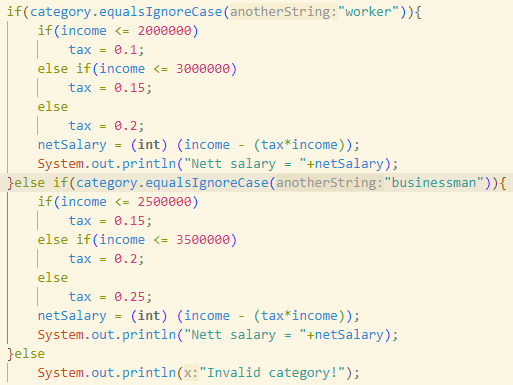
1. Create a new Java file named **Selection2Exp3StudentIDNumber.java**
2. Create a basic structure of Java program.
3. Import Scanner library
4. Declare an object of Scanner named **inputStudentIDNumber**
5. Declare variable **category**, **income**, **netSalary**, and **tax**.



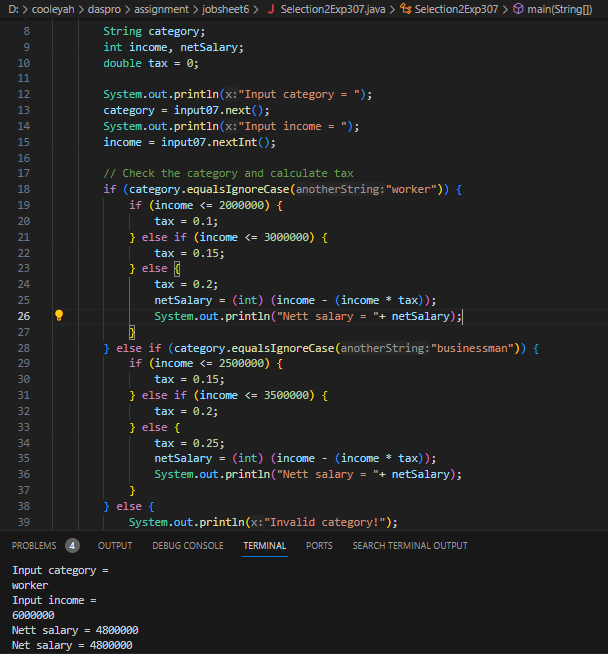
1. Add the following code to get user input.



1. Create a nested selection structure. The first condition is used to check the **category** (**worker** or **businessman**). Next, a second condition is carried out to determine the amount of **tax** based on the **income** that has been entered. Then add the code to calculate the **net**S**alary** received after tax deductions



1. Run and observe the result!

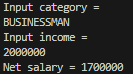


# Question

1. Explain the function of **(int)** in the syntax **netSalary= (int) (income - (income \* tax));**

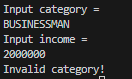
* (int) digunakan untuk mengubah hasil dari perhitungan (income – (income \* tax)) dari pecahan ke bilangan bulat (tanpa angka desimal). Jika hasil awal 2500.60 dibulatkan menjadi 2500 dengan menghilangkan angka desimalnya.

1. Run the program by entering **category** = **BUSINESSMAN** and **income** = **2000000**. Observe what happens! What is the use of **equalsIgnoreCase**?



* The **equalsIgnoreCase** method is used to compare strings for equality while ignoring their case. In the context of your program, it's used to compare the input category to the expected category values ("worker" or "businessman") without being case-sensitive.

1. Change **equalsIgnoreCase** to **equals**, then run the program by entering **category** = **BUSINESSMAN** and **income** = **2000000**. Observe what happens! Why is the result like that? What is the use of **equals**?



* equals is case sensitive, which means it treats uppercase and lowercase letters as different things. So, if use equals to compare "BUSINESSMAN" (uppercase) with "businessman" (lowercase), it will see them as different words and not consider them the same.

# 3. Assignment

**Experiment Time: 160 Minutes**

Create a program code based on the flowchart that was created in the 6th Fundamentals Programming Course meeting assignment related to the Project! Push and commit the results of your program code to your project repository!

**Note**: assignments may only apply material from meeting 1 to meeting 6.