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

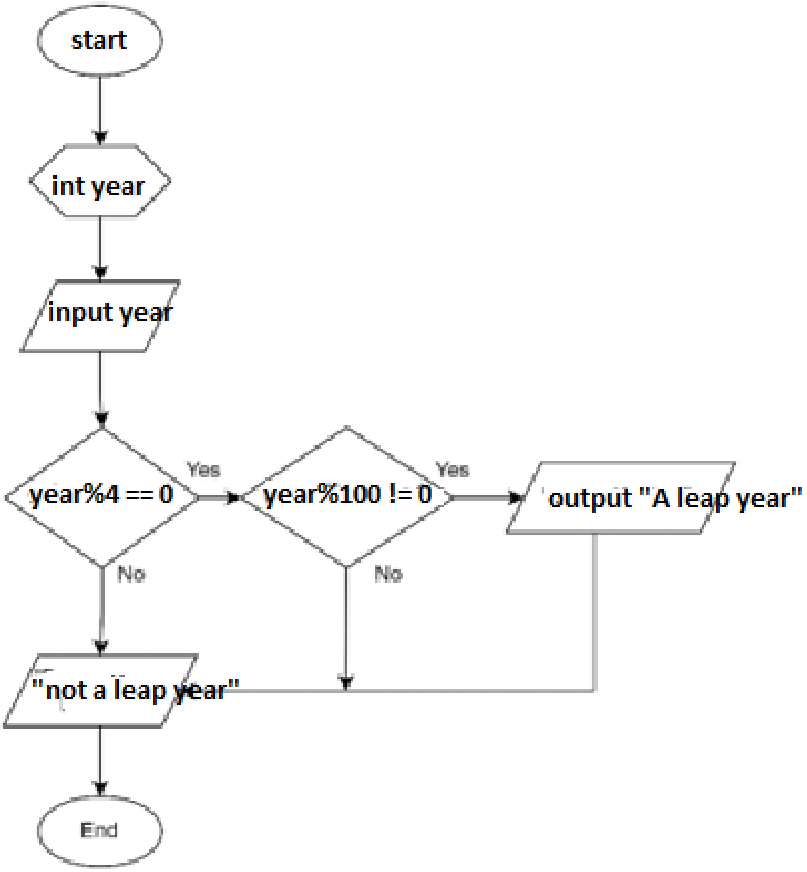
2. Laboratory

2.1 Experiment 1

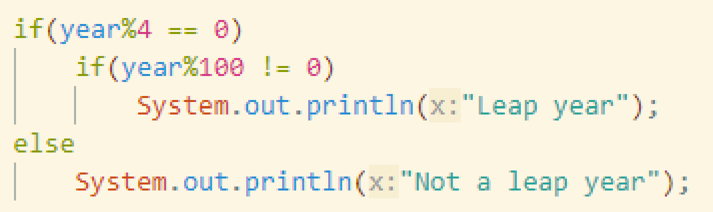
Experiment Time: 50 minutes

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

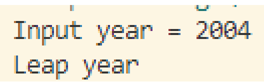


2. Open your project folder in VSCode, and create a new Java file named **Selection2Exp1StudentIDNumber.java**
3. Create a basic structure for Java program (including class and method main)
4. Add import statement fo Scanner library
5. Create a new object from Scanner named **inputStudentIDNumber**
6. Add a statement to get the user input.
7. Create a selection structure as follows:



```
if(year%4 == 0)
    if(year%100 != 0)
        System.out.println(x:"Leap year");
else
    System.out.println(x:"Not a leap year");
```

8. Run the program and observe the result:



```
Input year = 2004
Leap year
```

9. Commit and push the changes into your repository.

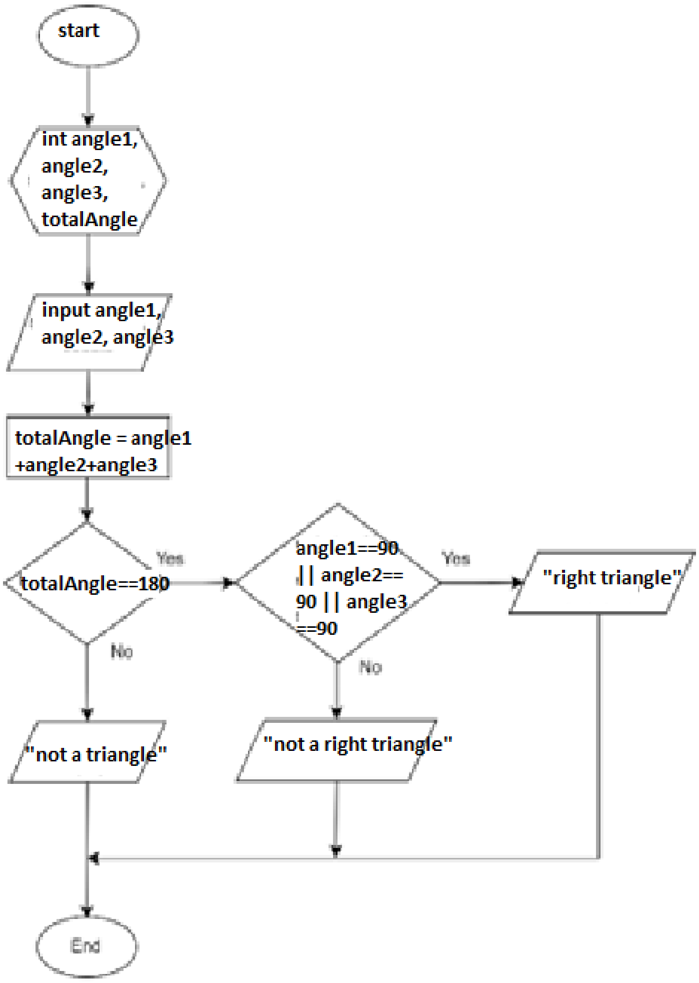
Question

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. Modify the program according to answer number 1!
3. **Commit and push the changes into your repository!**
4. 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)
5. Commit and push the modifications to the repository!

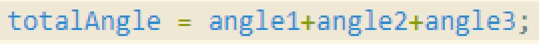
2.2 Experiment 2

Experiment Time: 50 minutes

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

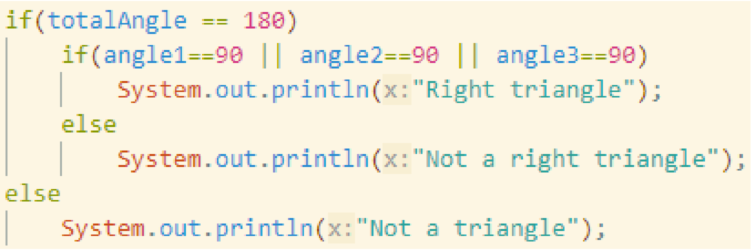

start
int angle1, angle2, angle3, totalAngle
input angle1, angle2, angle3
totalAngle = angle1 +angle2+angle3
totalAngle==180
Yes
No
angle1==90 || angle2== 90 || angle3 ==90
Yes
No
"right triangle"
"not a triangle"
"not a right triangle"
End

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



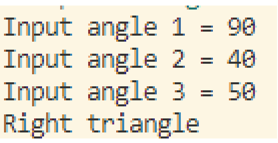
```java
totalAngle = angle1+angle2+angle3;
```

9. Create a selection structure as follows :



```java
if(totalAngle == 180)
    if(angle1==90 || angle2==90 || angle3==90)
        System.out.println(x:"Right triangle");
    else
        System.out.println(x:"Not a right triangle");
else
    System.out.println(x:"Not a triangle");
```

10. Run the program and observer the output:



```
Input angle 1 = 90
Input angle 2 = 40
Input angle 3 = 50
Right triangle
```

11. Commit and push the changes into your repository.

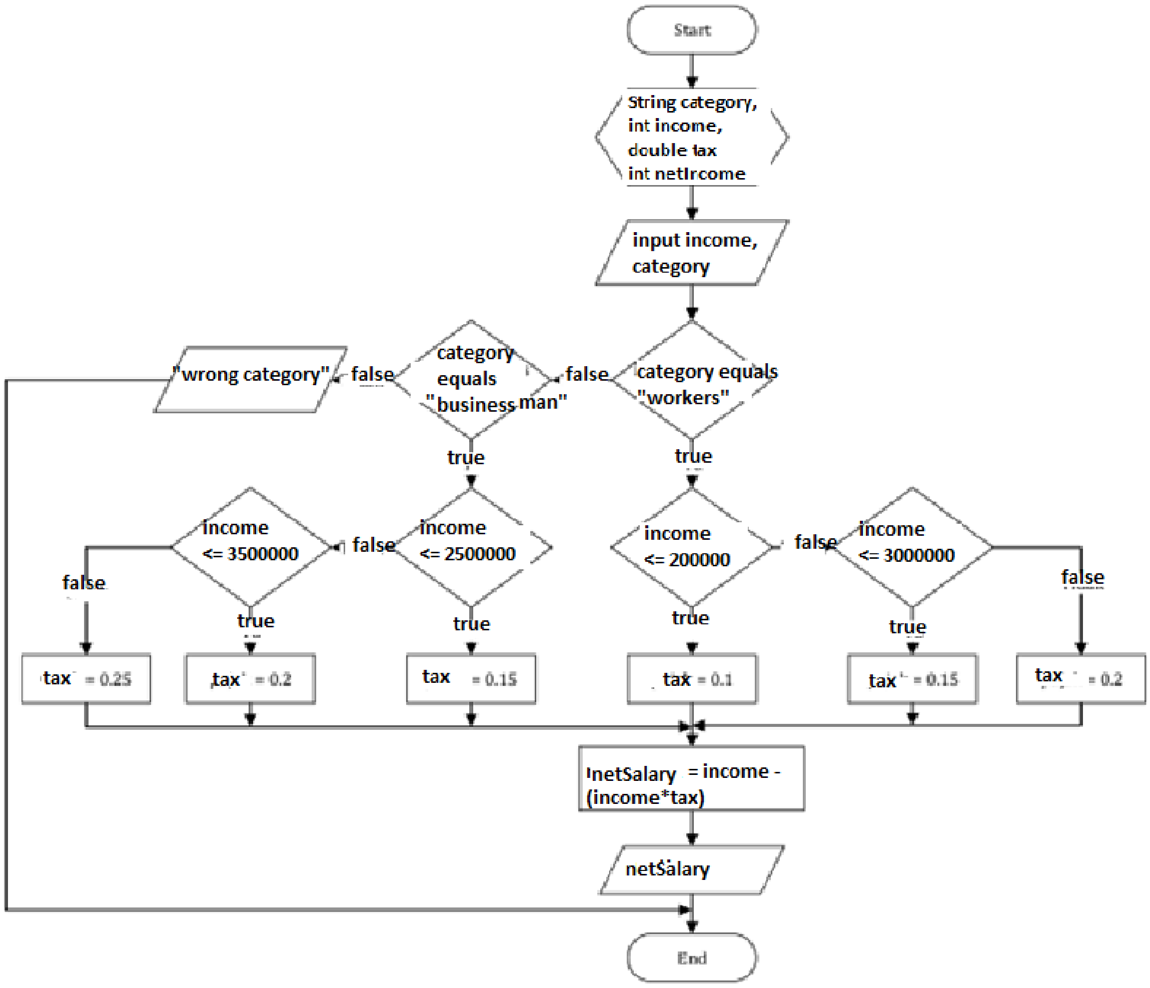
Question

1. Modify the source code, so that it can detect the other types of triangles (equilateral triangle and isosceles triangle)
2. Commit and push the changes into your repository.

2.3 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**.



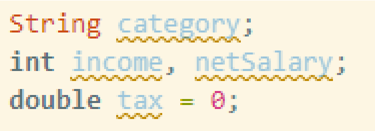
2. Create a new Java file named **Selection2Exp3StudentIDNumber.java**

3. Create a basic structure of Java program.

4. Import Scanner library

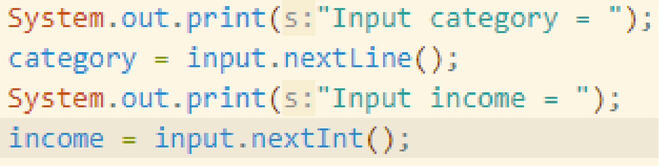
5. Declare an object of Scanner named **inputStudentIDNumber**

6. Declare variable **category**, **income**, **netSalary**, and **tax**.



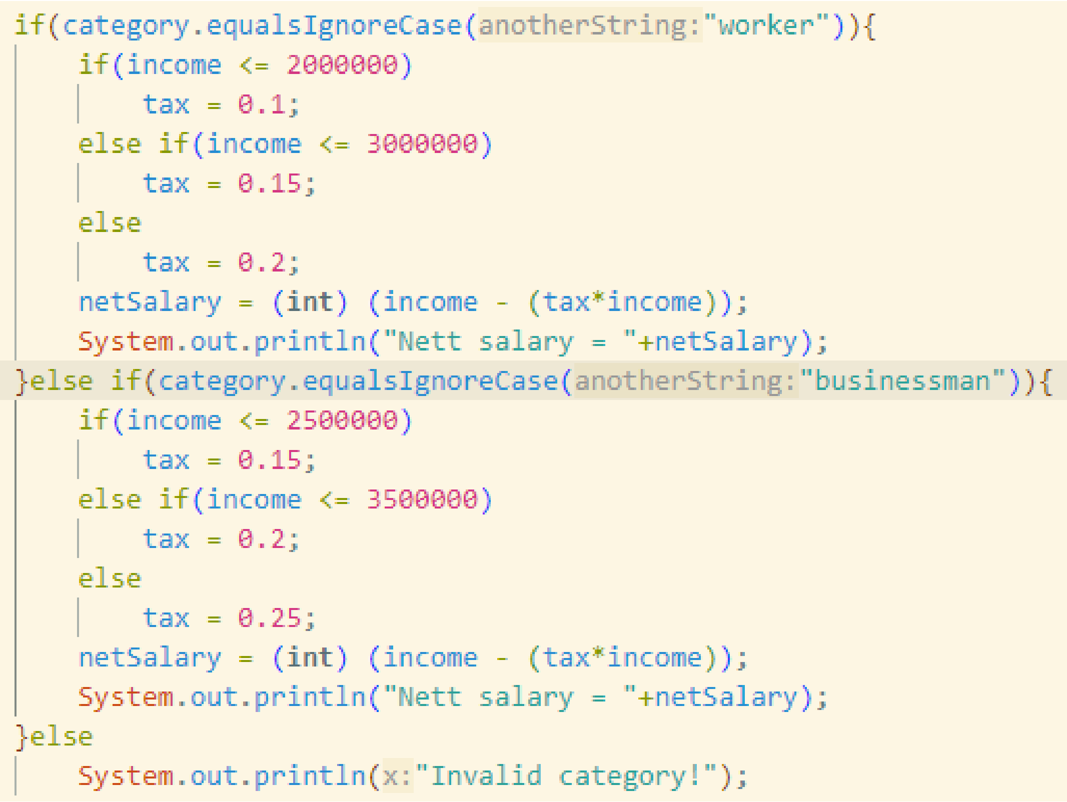
```
String category;
int income, netSalary;
double tax = 0;
```

7. Add the following code to get user input.



```
System.out.print(s:"Input category = ");
category = input.nextLine();
System.out.print(s:"Input income = ");
income = input.nextInt();
```

8. 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 **netSalary** received after tax deductions



```
if(category.equalsIgnoreCase(anotherString:"worker")){
    if(income <= 2000000)
        tax = 0.1;
    else if(income <= 3000000)
        tax = 0.15;
    else
        tax = 0.2;
    netSalary = (int) (income - (tax*income));
    System.out.println("Nett salary = "+netSalary);
}else if(category.equalsIgnoreCase(anotherString:"businessman")){
    if(income <= 2500000)
        tax = 0.15;
    else if(income <= 3500000)
        tax = 0.2;
    else
        tax = 0.25;
    netSalary = (int) (income - (tax*income));
    System.out.println("Nett salary = "+netSalary);
}else
    System.out.println(x:"Invalid category!");
```

9. Run and observe the result!

Question

1. Explain the function of **(int)** in the syntax **netSalary= (int) (income - (income * tax));**
2. Run the program by entering **category** = **BUSINESSMAN** and **income** = **2000000**. Observe what happens! What is the use of **equalsIgnoreCase**?
3. 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**?

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.