Midterm Laboratory Activity 1

Score:

Name: _____
Subject Code & Schedule: _____
Course and Year: _____

TITLE: if-else Statement

LEARNING OBJECTIVES:

At the end of this activity, the students should be able to:

1. Perform conditional statement using **if** and **if-else**.
2. Differentiate the function of an **if** and **if-else** statements.
3. Debug programs using nested **if** and cascading **if-else** statements.
4. Create a complete Java program that simulates these basic operations.

INSTRUCTIONS:

1. Make sure you have your own individual account.
2. Always keep your account secret to others to avoid unauthorized access to your files.
3. Always save your work and log-off when not using the computer.
4. By now you should have been familiarized using your text editor.
5. By now you should know how to create, save, compile, execute, and debug programs in Java.
6. Use the skills and learning obtained in Prelim Activity 1 to Prelim Activity 6 in order for you to successfully finish the learning objectives of this module.

DURATION: One to two Meetings

HANDS-ON:

1. Log-on using your own individual account. Use your own **username** and **password.**
2. Open your text editor.
3. Write your next Java program.
   3.1. Write your next program by copying the source code shown below to your text editor.

```
/* Programmed by: <write your name here>
   Program title: PosNeg.java */
   Program Date: <write the date today here> */

import java.io.*;
public class PosNeg{
  public static void main(String[] args){
    int number;
    String input = " ";

    BufferedReader in = new BufferedReader(new
                        InputStreamReader(System.in));

    System.out.println("Input a Number: " );

    try{
      input = in.readLine();
    }catch(IOException e){
      System.out,println("Error!");
    }

     number = Integer.parseInt(input);

     if(number < 0)
     {
       System.out.println("The number " + number + " is
                           NEGATIVE");
     }
     if(number >= 0)
     {
       System.out.println("The number " + number + " is
                           POSITIVE");
     }
   }
}
```

3.2. Save your program as **PosNeg.java** then compile your program until no errors and warnings are reported.

3.3. Run your program.

3.4. Simulate and write what will be displayed on the screen.

4. Create a new program and save it as **`TwoNum.java.`** Your program should allow the user to input two integer numbers. Thereafter, the program should determine if these two integer numbers are **EQUIVALENT**, or if the first number is **GREATER THAN** the second number, or if the first number is **LESSER THAN** the second number.

4.1. Write your complete program here:

4.2. Save then compile your program until no errors and warnings are reported.
4.3. Run your program.
4.4. Simulate and write what will be displayed on the screen.

5. Rewrite your **`PosNeg.java`** source code using only **one if statement** and an **else statement**. Save it as **`PosNeg2.java`**

5.1. Write your complete source code here:

5.2. Save then compile your program until no errors and warnings are reported.
5.3. Run your program.
5.4. Write what will be displayed on the screen after a complete simulation of your program.

6. Rewrite your **`TwoNum.java`** source code and save it as **`TwoNum2.java`** but this time using cascading **if-else** statements.

   6.1. Write your complete source code here:

6.2. Save then compile your program until no errors and warnings are reported.
6.3. Run your program.
6.4. Write what will be displayed on the screen after a complete simulation of your program.

7.1. Complete the source code below:

```
int age;
String input = " ";

BufferedReader in = new BufferedReader(new
                    InputStreamReader(System.in));

System.out.print("Input age : ");

try{
  input = in.readLine();
}catch(IOException e){
  System.out.println("Error!");
}

age = Integer.parseInt(input);

if(age < 0 || age >= 100)
     System.out.print("Invalid age");
if(age < 5)
     System.out.println("BABY");
if(age < 12)
     System.out.println("CHILD");
if(age < 20)
     System.out.println("TEENAGER");
if(age < 50)
     System.out.println("ADULT");
if(age < 100)
     System.out.println("SENIOR CITIZEN");
```

7.2. Save your program as **`Age.java`** then compile your program until no errors and warnings are reported.
7.3. Run your program.
7.4. Write what will be displayed on the screen for the following input.

```
Input age : 101
```

```
Input age : 95
```

```
Input age : 50
```

```
Input age : 45
```

```
Input age : 13
```

```
Input age : 10
```

```
Input age : 3
```

```
Input age : -2
```

7.5. Based from the results of your simulation, write down what you have observed.

_____
_____
_____
_____
_____
_____
_____
_____

8. Now try revising the **`Age.java`** program by using cascading if-else statement instead of the nested if-statements.

8.1. Write your revised cascading if-else statement here:

```
```

8.2. Simulate using the same test inputs. Write what will be displayed on the screen:

```
Input age : 101
```

```
Input age : 95
```

```
Input age : 50
```

```
Input age : 45
```

```
Input age : 13
```

```
Input age : 10
```

```
Input age : 3
```

```
Input age : -2
```

8.3. Based from the results of your simulation, write down what you have observed.

_____
_____
_____
_____
_____
_____
_____
_____

8.4. Is there any difference between a nested if and cascading if-else statements? If so, differentiate it.

_____
_____
_____
_____
_____
_____

9. Create a new program and save it as **`Bingo.java`**

9.1. You were hired by PAGCOR as part of the programming team in-charge of automating its BINGO game. Your task is to write a program that will accept an integer number whose value is from 1 to 75. Thereafter, your program should determine and print the letter that corresponds to that number. That is, numbers 1 to 15 corresponds to the letter 'B', 16 to 30 corresponds to the letter 'I', 31 to 45 corresponds to the letter 'N', 46 to 60 corresponds to the letter 'G' and 61 to 75 corresponds to the letter 'O'.

9.2. Write your complete source code here

9.3. Save then compile your program until no errors and warnings are reported.
9.4. Run your program.
9.5. Write what will be displayed on the screen during your simulation or test run.