National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS 212: Object Oriented Programming

Class: BESE-7AB

Lab02: Control Structures

Date: February 24, 2017

Instructor:

Dr. Muhammad Muneeb Ullah

Learning Objectives

The objective is to comprehend and practice Java's Control Structures, such as equality and relational operators, if…else selection, and loops.

Activity #1

Your first activity will be to test the `if` statements, relational and equality operators in Java. Test the following program:

```
// Compare integers using if statements, relational operators
// and equality operators.
import java.util.Scanner; // program uses the class Scanner

public class Comparison
{
      // main method begins execution of Java application
      public static void main( String[] args )
      {
            // create Scanner to obtain input from command line
            Scanner input = new Scanner( System.in );

            int number1; // first number to compare
            int number2; // second number to compare

            System.out.print( "Enter first integer: " ); // prompt
            number1 = input.nextInt(); // read first number from user

            System.out.print( "Enter second integer: " ); // prompt
            number2 = input.nextInt(); // read second number from user

            if ( number1 == number2 )
                  System.out.printf( "%d == %d\n", number1, number2 );

            if ( number1 != number2 )
                  System.out.printf( "%d != %d\n", number1, number2 );

            if ( number1 < number2 )
                  System.out.printf( "%d < %d\n", number1, number2 );

            if ( number1 > number2 )
                  System.out.printf( "%d > %d\n", number1, number2 );

            if ( number1 <= number2 )
                  System.out.printf( "%d <= %d\n", number1, number2 );

            if ( number1 >= number2 )
                  System.out.printf( "%d >= %d\n", number1, number2 );
```

```
      } // end method main
} // end class Comparison
```

Activity #2

Transform the above program to use nested `if…else` double-selection structures, and make sure that you get the same results.

Activity #3

There is a mysterious error in this program. Identify this error. Is this a syntax error or logical? Moreover, correct the error.

```
public class Calculate
{
      public static void main(String[] args)
      {
            int sum;
            int x;
            x = 1;      // initialize x to 1 for counting
            sum = 0;    // initialize sum to 0 for totaling

            System.out.printf( "Going to calculate the sum" );

            while ( x <= 10 ) // while x is less than or equal to 10
            {
                  sum += x;  // add x to sum
            } // end while

            System.out.printf( "The sum is: %d\n", sum );
      }
}
```

Activity #4

What does the following program print? Predict the output, before you actually try this program.

```
public class Mystery
{
      public static void main( String[] args )
      {
            int y;
            int x = 1;
            int total = 0;

            while ( x <= 10 )
            {
                  y = x * x;
                  System.out.println( y );
                  total += y;
                  ++x;
            } // end while

            System.out.printf( "Total is %d\n", total );
      } // end main
} // end class Mystery
```

Task #1

The formula for calculating Body Mass Index (BMI) is:

$$\text{BMI} = \frac{Weight}{Height * Height}$$

Create a BMI calculator that reads the user's weight in kilograms and height in meters, then calculates and displays the user's body mass index.

Further, with the help of the following information, categorize the user's BMI into one of the categories: {Underweight, Normal, Overweight, Obese}.

BMI VALUES

Underweight: less than 18.5
Normal: between 18.5 and 24.9
Overweight: between 25 and 29.9
Obese: 30 or greater

Task #2

Drivers are concerned with the mileage obtained by their automobiles. One driver has kept track of several tankfuls of petrol by recording miles driven and liters used for each tankful.

Develop a Java application that will input the miles driven and liters used for each tankful. The program should calculate and display the miles per liter obtained for each tankful. After processing all input information, the program should calculate and print the combined miles per liter obtained for all tankfuls. Here is a sample input/output dialog:

```
Enter the liters used (-1 to end): 12.8
Enter the miles driven: 287
The miles/liter for this tank was 22.421875

Enter the liters used (-1 to end): 10.3
Enter the miles driven: 200
The miles/liter for this tank was 19.417475

Enter the liters used (-1 to end): 5
Enter the miles driven: 120
The miles/liter for this tank was 24.000000

Enter the liters used (-1 to end): -1

The overall average miles/liter was 21.601423
```

Task #3

Write a Java application that inputs an integer containing only 0s and 1s (i.e., a binary integer) and prints its decimal equivalent. [*Hint*: Use the remainder and division operators to pick off the binary number's digits one at a time, from right to left].

In the decimal number system, the rightmost digit has a positional value of 1 and the next digit to the left a positional value of 10, then 100, then 1000, and so on. The decimal number 234 can be interpreted as 4 * 1 + 3 * 10 + 2 * 100.

In the binary number system, the rightmost digit has a positional value of 1, the next digit to the left a positional value of 2, then 4, then 8, and so on. The decimal equivalent of binary 1101 is 1 * 1 + 0 * 2 + 1 * 4 + 1 * 8, or 1 + 0 + 4 + 8 or, 13.

Hand in

Hand in the source code from this lab at the appropriate location on the LMS system. You should hand in a single compressed/archived file named Lab_2_<Your CMS_ID. Your_NAME >.zip (without angle brackets) that contains ONLY the following files.

1) All completed Java source files representing the work accomplished for this lab: ActivityOne.java; ActivityTwo.java; ActivityThree.java; ActivityFour.java; Task1.java, Task2.java, and Task3.java. The files should contain author name in the comments at the top.

2) A plain text file named **README.TXT** that includes a) author information at the beginning, b) a brief explanation of the lab activities/tasks, and c) any comments, or suggestions.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.

2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.

3. The lab time is not intended as free time for working on your programming/other assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the programming/other assignments.