Tutorial 02: Introduction to Problem Solving, Selection Control Statements & Logical Operators

Exercise 1

Write a Java program that computes the maximum of 3 numbers entered by the user.

```
import java.util.Scanner;


public class Exercise01
{
   public static void main(String[] argv)
   {
      // declare a scanner to input numbers
      Scanner input = new Scanner(System.in);

      // declare 3 integer variables to input
      int number1, number2, number3;

      // declare the max value to find
      int max;

      // prompt the user to enter 3 integers
      System.out.println("Enter three numbers: ");
      number1 = input.nextInt();
      number2 = input.nextInt();
      number3 = input.nextInt();

       // assign the first variable value to max
      max = number1;

      // compare the max with the other variables:
```

```
        //  - if the max still bigger, keep its value as it is
        //  - if the max is smaller, update its value

        // compare the max with number2 and update the max if needed
        if ( max < number2 )
            max = number2;

        // compare the max with number3 and update the max if needed
        if ( max < number3 )
            max = number3;

        // print the max value
        System.out.printf("Max = %d\n", max);
    }
}
```

Exercise 2 (textbook)

Write a Java application that reads two integers, determines whether the first is a multiple of the second and prints the result.

```
import apple.laf.JRSUIConstants;
import java.util.Scanner;
public class Exercise02 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int number1, number2;

        System.out.print("Enter two numbers: ");
        number1 = input.nextInt();
        number2 = input.nextInt();

        if (number1 % number2 == 0)
```

```
            System.out.printf("%d is multiple of %d\n", number1, number2);


        if (number1 % number2 != 0)
            System.out.printf("%d is not multiple of %d\n", number1, number2);


    }
}
```

Problem (past exam)

(`class ConvertHours`) Write a Java application that takes a positive integer representing a duration in hours as input, and outputs the same duration decomposed into weeks, days, and hours as detailed below.

unit	suffix used in output	conversion
week	wk	1 week = 7 days
day	d	1 day = 24 hours
hour	hr	

You must consider the following:

- **Accept only strictly positive value of hour and less than or equal to 8760**
- **Only include quantities with non-zero values in the output (e.g., print "1 d" and not "0 wk, 1 d, 0 hr ").**
- **Give larger units precedence over smaller ones as much as possible (e.g., print 2 d, 10 hr and not 1 d, 34 hr or 58 hr).**

```
package imamu.ccis.cs.cs140.tutorials.tutorial02;


import java.util.Scanner;


public class ConvertHours
{


    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
```

```
System.out.print("Enter number of hours: ");
int hours = input.nextInt();

// check if the number of hours is correct
if ( hours >= 1 && hours <= 8760)
{
   int weeks; // compute the number of weeks
   weeks = hours / (24 * 7);
   hours %= (24 * 7);

   int days; // compute the number of days
   days = hours / 24;
   hours %= 24;

   if (weeks >= 1)
      System.out.print(weeks + " wk ");

   if (days >= 1)
      System.out.println(days + " d ");

   //print the number of hours if it remains more than one hour
   if ( hours >= 1)
      System.out.print(hours + " hr ");

   System.out.println();
} else {
   System.out.println("Incorrect number of hours.");
}
```

```
    }
}
```