CMT205 Object Oriented Development with Java

Lab Exercises Week 2

Section 1: Arithmetic and Decisions

1. Use nested **if** statements to find the longest length of 3 sides of a triangle.

   Note that you give the values of variables at the beginning of your program. We will practise how to get input from the user later on.

   Hint: it is often easier to start describing your algorithm (solution) using some pseudo code (like Java code but with natural language when needed) and then covert your algorithm to the Java code. This is more important when the problem becomes more complicated.

Example solution:

```
public class Longest
{
   // Longest side of a triangle
   public static void main(String[] args)
   {
      int a = 5;
      int b = 3;
      int c = 4;
      int longest;
      if ( a > b )
      {
         if ( a > c )
            longest = a;
         else
            longest = c;
      }
      else
      {
         if ( b > c )
            longest = b;
         else
            longest = c;
      }
      System.out.println( "Longest side is " + longest );
   }
}
```

2. A leap year is a year containing one additional day. Given a year in the Gregorian calendar, print whether the year is a leap year or not.

   Hint: The rules to determine if a given year is a leap year are as follows:
   a. if the year is not divisible by 100, then it is a leap year if and only if the year is divisible by 4.
   b. if the year is divisible by 100, then it is a leap year if and only if it is divisible by 400

   So 2000 is a leap year (divisible by 400), while 2100 is not (divisible by 100 but not 400).

3. Convert an `int` number `weekday` to the weekday name `weekdayName`. So 0 is mapped to "Sunday", 1 is mapped to "Monday" … 6 is mapped to "Saturday" and an error message is printed for `weekday` out of the range of 0 to 6.

Hint: Use `switch` to check different conditions.

Note: We will later find a simpler way of achieving this by using an array.

Section 2: Loops

1. Write a **for** loop to add the numbers 1 to 10.

Example Solution:

```
public class Addition
{
   public static void main( String [] args )
   {
      int total = 0;
      for( int value = 1; value <= 10; value++ )
      {
         total = total + value;
      }
      System.out.println( "Total = " + total );
   }
}
```

2. Write a **for** loop to perform the sum of the following series:
   - 2+4+6+8+10+12+14+16

3. Write a **while** loop to perform the sum of the following series:
   - 2+4+6+8+10+12+14+16

4. Write a program which checks whether a number is a prime number (i.e. a number only divisible by itself and one.) The number one is not a prime number. The number to be checked is specified in the beginning of the program.
   Hint: Use `a%b==0` to test if `a` is divisible by `b`.

Section 3: Keyboard and Command Line Input

1. Following the Question 4 in Section 2, after the program starts, prompt "Enter number >>" and ask the user to enter a number to be checked. Then, the program outputs whether the given number is (or is not) a prime number.

Hint: use **String.parseInt** static method to convert a String to an integer.

2. Following the previous question, modify the program such that it keeps asking the user to enter numbers, until the user types "QUIT". For every input number, check if the number is a prime number and output the corresponding message.

3. Write a program to check if a given number is a prime number. The program takes one command line argument. If there are no or more than one command line arguments, then an error message should be printed. Otherwise, check if the number is a prime number and output the corresponding message.