C212/A592 Spring 17 Lab 3

Intro to Software Systems

Objectives:
- Implementing method
- Reading values from keyboard (Using Scanner class)
- Using a simple loop
- Casting

Lab instructions

1. Create a class named Lab3Exercises
2. Implement the following static methods (we will not use this class to *instantiate objects*)
   a. *public static int fib()*
      i. Using a while loop return the nth number in the Fibonacci sequence. Here is a list of the Fibonacci numbers
      https://en.wikipedia.org/wiki/Fibonacci_number#List_of_Fibonacci_numbers
      ii. Use the *Scanner* class covered in lecture to get input from the user rather than passing a parameter to the method.

   b. *public static String numbers()*
      i. Using the *Scanner* class and a while loop, continue to process integer values from a user until they have input a value other than an *int.*
      ii. Return a *String* with the sum of all the numbers, the min value, and the max value
      iii. Do not use Java's *min* or *max* methods

   c. *public static String grade()*
      i. Using *the Scanner* class, process an integer value between [0 – 100]. Return a String value of the letter grade.
      ii. e.g., input = 85 return *"Your grade is a B"*
      iii. *grade greater than 89.4 –A*
      iv. *grade greater than 79.4 –B*
      v. *grade greater than 69.4 –C*
      vi. *grade greater than 59.4 –D*
      vii. *grade less than 59.4 –F*

   d. *public static String intToBinary(int n)*
      i. Implement a method that puts the binary representation of a positive *integer N* into a *String S.* You do not need to have leading zeroes, e.g., 1 in binary is 1, 2 in binary is 10, 3 in binary is 11

ii. Note: Do not use the java method *Integer.toBinaryString(N)*

iii. Here are two very good explanations to converting an integer value to binary http://www.wikihow.com/Convert-from-Decimal-to-Binary

iv. Examples:
intToBinary(4) returns 100
intToBinary(8) returns 1000

e. Answer the following questions as comments below your class file:
   i. Give the type and value for each of the following expressions:
      1. *(1 + 2.236)/2*
      2. *1 + 2 + 3 + 4.0*
      3. *4.1 >= 40*
      4. *1 + 2 + "3"*

   ii. Give the value printed by each of the following code fragments:
      1. The following code in 1 is using Newton's Method to calculate the square root of a number

         ```
         double t = 9.0;
         while (Math.abs(t – 9.0/t) > .001) {
                 t = (9.0/t + t) / 2.0;
         }
         System.out.println(t);
         ```

      2. ```
         int sum = 0;
         for (int i = 1; i < 1000; i++) {
                 for (int j = 0; j < 1000; j++) {
                         sum++;
                 }
         }
         System.out.println(sum);
         ```

      3. ```
         int sum = 0;
         for (int i = 1; i < 1000; i *= 2) {
                 for (int j = 0; j < 1000; j++) {
                         sum++;
                 }
         }
         System.out.println(sum);
         ```