CSCI-C212/A592 – Intro to Software Systems
Spring 2017

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

Assignment 2

Due by 2/10/2017, Friday midnight through Canvas

Instructions:

- Review the requirements given below and complete ***all three*** parts from below. Please submit all files through Canvas.
- The grading scheme is provided on Canvas. Be sure that your code includes everything required in the grading rubric.

Part 1:

1. Create a class that will have the following static methods as exercises.
   a. *public static boolean isPalindrome(String s)*
      i. Checks whether a string is a palindrome. (A word that reads the same backwards and forwards e.g., racecar)
      ii. You cannot reverse the string with built in reverse method in Java.

   b. *Public static boolean isSorted(String s)*
      i. Checks whether a string is in alphabetical order. Each letter in the string must not precede any letters that appear before it in the alphabet.
      e.g., bdfjk is in alphabetical order
      bdcfjk is not
      ii. It would be a good idea to look at the String and Character API in the Java docs for what methods to use. Specifically that Character.compare() methods, and combine this with the String charAt() method.

   c. *public static void border(int n)*
      i. Using a double nested for loop, print a *nxn* square where the border is filled with asterisks and the center is blank.
      (the spacing is not symmetrical in MS Word with line vs space differences but it will be when just printing asterisks and spaces when appropriate in your Java program)

```
* * * * *
*       *
*       *
*       *
* * * * *
```

   d. *public static void T(int n)*
      i. Using a double nested for loop, print a T of asterisks.

Part 2: Dice Simulation

Develop a dice rolling simulation to collect statistics of rolling a die a large number of times.

2. Create a class named Counter with the following API

   private final String name; // instance field
   private int count // instance field

   public Counter(String id) // Counter constructor – Notice Constructors do not have a return type and are named the same as their class

   public void increment() // increments count

   public int tally() // returns count

   public int toString() // prints name and count

3. Create a class Die with the following API

   private Random rand; // instance field

   public Die() // initialize a new Random object in the Constructor

   public int roll() // returns a random number between [1,6]
   // Look at the nextInt(int bound) method of Random

4. Create a Class DieSimulation with the following API

   // instance fields
   private Die die;
   private Counter one;
   private Counter two;
   private Counter three;
   private Counter four;
   private Counter five;
   private Counter six;

   public DieSimulation() // initialize all the instance fields

   public String run()

a. In the run method, use a Scanner to receive input from the user for how many times to roll the die.
b. Using the Counter objects, record the outcome of each roll.
c. Return all the counts of all the Counters

Part 3:

- Write a program to simulate a bank transaction. There are two bank accounts: checking and savings.
  First, ask for the initial balance of the bank accounts, rejecting negative balances. Then display a menu (typical menu you see on an ATM machine without the graphics) and prompt the user to select the type of transaction (1: Withdrawal, 2 for Deposit, 3 for Balance, or 4 for exit). Then ask for the account type (checking or savings). Reject transactions that overdraw an account.
  Display starting and ending balance for each transaction. You program must never end unless user chooses exit option from the menu.