SI@UCF Java Course Test #1 Solutions June 14, 2019

1. (12 pts) Write a program that simulates a two player game with dice. Both players are to roll a pair of fair standard six-sided dice. Print out the value shown on each die for both players. The winner is the person who rolled a higher sum. Print out who the winner is, or if both players rolled the same sum, print out that the game was a tie. Please use the variables given to you, and you can also define other variables to use.

```
import java.util.*;
public class dicegame {
    public static void main(String[] args) {
        Random randGen = new Random();
        int p1Die1, p1Die2, p2Die1, p2Die2;
          plDie1 = 1 + randGen.nextInt(6);
          plDie2 = 1 + randGen.nextInt(6);
          p2Die1 = 1 + randGen.nextInt(6);
          p2Die2 = 1 + randGen.nextInt(6);
          int plScore = plDie1 + plDie2;
          int p2Score = p2Die1 + p2Die2;
          if (p1Score > p2Score)
               System.out.println("Player 1 won!");
          else if (p2Score > p1Score)
               System.out.println("Player 2 won!");
          else
               System.out.println("Game was a tie.");
    }
}
Grading: 4 pts for dice generation, 2 pts for totaling dice,
         2 pts for each branch of the if
```

2. (8 pts) Let the variable x be a double that stores an angle (in radians). Write a line of code to print out the value of $e^{\cos(x)}$ using the appropriate methods from the Math library (list included at the end of the exam).

System.out.println(Math.exp(Math.cos(x)));

Grading: 2 pts print, 2 pts Math.exp, 2 pts Math.cos, 2 pts x

3. (15 pts) Write a static method called shuffle that takes in two String objects str1 and str2, both guaranteed to be of even length, splits both of them into two halves (left and right), and returns the string that is the concatenation of the left half of the first string, followed by the left half of the second string, followed by the right half of the first string and then followed by the right half of the second string. For example if str1="computer" and str2="programs", then your method should return the string "compproguterrams". Please fill in the method given below:

```
String str1Left = str1.substring(0, str1.length()/2);
String str1Right = str1.substring(str1.length()/2);
String str2Left = str2.substring(0, str2.length()/2);
String str2Right = str2.substring(str2.length()/2);
return str1Left + str2Left + str1Right + str2Right;
```

}

Grading: 3 pts for each substring, 3 pts to paste back together.

Consider creating a class to store a domino. A domino is a playing piece that contains a certain number of dots on one side (in between 1 and 9 inclusive), and another number of dots on the other side. For clarity, we'll call the sides left and right. An incomplete version of the class is included below. The next four problems (5 - 8) will be to add methods to this class, and question 9 will ask you to use the class.

```
public class Domino {
    private int left;
    private int right;

    public Domino(int 1, int r) {
        left = 1;
        right = r;
    }

    public Domino(Random r) {
        left = Math.abs(r.nextInt()%9) + 1;
        right = Math.abs(r.nextInt()%9) + 1;
    }

    public String toString() {
        return " ["+left+", "+right+"]";
    }
}
```

4) (10 pts) Add a constructor to the Domino class that takes in a single integer and sets both the left and right side to have that many dots.

```
public Domino(int dots) {
    left = dots; // Grading: 5 pts each line, Give 3 pts for a
    right = dots; line that is backwards, 0 otherwise
}
```

5) (10 pts) Add a method flip to the Domino class that changes the current object by exchanging the number of dots on the left and right side. For example, if the object the method is called upon has 3 dots on the left and 5 on the right before the method call, right after the method call the object would have 5 dots on the left and 3 on the right.

```
public void flip() {
    int temp = left; // 4 pts
    left = right; // 3 pts
    right = temp; // 3 pts
}
```

6) (10 pts) Add a method compareTo to the Domino class so that it returns an integer. In particular it will return the difference in the total number of dots on the current object with the object d, which is passed in as a parameter. (Thus, a negative integer is returned if the current object has fewer dots, 0 is returned if they have the same number of dots, and a positive integer is returned if the current object has more dots than d.

```
public int compareTo(Domino d) {
    return this.left + this.right - d.left - d.right;
    // Grading: 2 pts return 2 pts each term and sign together
}
```

7) (15 pts) Add a method match to the Domino class and determines whether or not the current object matches a Domino d. Two dominos match if the number of dots on one side of the first domino is equal to the number of dots on one side of the second domino. (For example, the domino [2,7] matches [7,3] and the domino [1,5] matches [1,8], but domino [4,5] does NOT match [3,6].)

```
public boolean match(Domino d) {
```

}

```
// 5 pts for first two cases.
if (this.left == d.left || this.left == d.right)
  return true;
// 5 pts for next two cases.
if (this.right == d.left || this.right == d.right)
  return true;
// 5 pts for false case.
return false;
```

8) (15 pts) Write a code segment in the draw function to create a set of 13 horizontal stripes that are alternating between yellow and black, starting with yellow. The stripes should be 400 pixels long and 20 pixels wide. The Graphics class methods and Color class constants that are necessary are provided below. (Note: x is measured to the right and y is measured down.)

 abstract void
 fillRect(int x, int y, int width, int height)

 Fills the specified rectangle.

 abstract void
 setColor(Color c)

 Sets this graphics context's current color to the specified color.

 static Color
 YELLOW

 The color yellow.

 static Color
 BLACK

 The color black.

public void draw(Graphics g) {
 g.setColor(Color.WHITE);
 g.fillRect(0, 0, 500, 500);
 int y = 120;
 for (int i=0; i<13; i++) { // 2 pts
 if (i%2 == 0) // 3 pts
 g.setColor(Color.YELLOW); // 1 pt
 else // 1 pt
 g.setColor(Color.BLACK); // 1 pt
 g.fillRect(50, y, 400, 20); // 5 pts</pre>

```
g.fillRect(50, y, 400, 20); // 5 pts
y += 20; // 2 pts
```

}

}

9) (5 pts) Food from what country is served at Café de France? France (5 pts give to all)