COSC 117 **Exam #2 Solutions** Spring 2017

1 Short Answer (10 Points Each)

1. Write a method that takes in an integer n as a parameter and returns a long that is the value of $n!$. Recall that $n! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1$.

   Solution:

```java
public static long fact(int n) {
    long nfact = 1;
    for (int i = 1; i <= n; i++) {
        nfact *= i;
    }
    return nfact;
}
```

2. Write a method that will take two integers as parameters, n and t. The method should simulate the rolling of two die and count the number of times the total of the roll equals t. This rolling of the two die should be done n times and the number of times the sum is equal to t is to be returned as an integer. If t is not in the range of 2 to 12 the method should return -1.

   Solution:

```java
public static int rollcount(int t, int n) {
    if (t < 2 || t > 12)
        return -1;

    int count = 0;
    for (int i = 0; i < n; i++) {
        int r1 = (int) (Math.random() * 6 + 1);
        int r2 = (int) (Math.random() * 6 + 1);
        if (r1 + r2 == t)
            count++;
    }
    return count;
}
```

3. Write a method that takes in doubles x1, y1, x2, and y2 as parameters and returns the distance between the points (x_1, y_1) and (x_2, y_2). Recall that the distance between two points is calculated as $d = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$.

   Solution:

```java
public static double distance(double x1, double y1, double x2, double y2) {
    return Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
}
```

4. Write a method that takes in a single integer n and prints out n rows of asterisks. The first row has one, the second two, up to the last row that will have n. So if 7 is sent to this method the output would be the following.

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

   Solution:

```java
public static void method4(int n) {
    for (int i = 1; i <= n; i++){
        for (int j = 1; j <= i; j++){
            System.out.print("*");
        }
        System.out.println();
    }
}
```

5. Write a method that takes in as parameters a string and a character. The method should return the count of the number of occurrences of the character in the string. This is to be case-insensitive, so a and A would both be counted as an a.

   Solution:

```java
public static int charcount(String str, char c) {
    int count = 0;
    str = str.toLowerCase();
    c = Character.toLowerCase(c);
    for (int i = 0; i < str.length(); i++){
        if (str.charAt(i) == c)
            count++;
    }
    return count;
}
```

6. Write a method that takes in a string and counts the number of words in the string and returns the number of words. You may assume that every word is ended by a character that is not a letter.

   Solution:

```java
public static int wordcount(String str) {
    char lastchar = str.charAt(0);
    int count = 0;

    for (int i = 1; i < str.length(); i++){
        if (!Character.isLetter(str.charAt(i)) && Character.isLetter(lastchar))
            count++;
        lastchar = str.charAt(i);
    }
    return count;
}
```

2 Program Traces (15 Points Each)

1. For the given input, write the output of the program in the Program Output box.

```java
import java.util.Scanner;

public class Traces001 {

    public static int method1(int a, int b) {
        if (a < b) {
            b = b % a;
            System.out.println(b);
            a++;
        }
        return b;
    }

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

        System.out.print("-> ");
        int n = kb.nextInt();
        System.out.print("-> ");
        int t = kb.nextInt();

        int i = 1;
        System.out.println(i + "  " + n + "  " + t);

        while (n > 0) {
            i++;
            n = method1(n, t);
            System.out.println(i + "  " + n + "  " + t);
            t = t / 2;
        }
        System.out.println(i + "  " + t);
    }
}
```

Solution:

```
-> 6
-> 15

1  6  15
3
2  3  15
1
3  1  7
0
4  0  3
4  1
```

```
-> 5
-> 27

1  5  27
2
2  2  27
1
3  1  13
0
4  0  6
4  3
```

2. For the given input, write the output of the program in the Program Output box. Make sure that you represent spaces by our special symbol ␣.

```
import java.util.Scanner;

public class Trace002 {

    public static int method1(String s1, String s2) {
        int n = s1.lastIndexOf(s2);
        return method2(n);
    }

    public static int method2(int a) {
        if (a < 0) {
            return 0;
        } else if (a % 2 == 0) {
            return a / 2;
        } else {
            return 3 * a + 1;
        }
    }

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

        System.out.print("-> ");
        String str1 = kb.nextLine();
        System.out.print("-> ");
        String str2 = kb.nextLine();

        int n = str1.length();
        while (n > 1) {
            n = method1(str1, str2);
            if (n > str1.length())
                n = str1.length() - 1;
            str1 = str1.substring(0, n);
            System.out.println(n);
            System.out.println(str1);
        }
    }
}
```

Solution:

```
-> This␣is␣a␣stupid␣test
-> s
20
This␣is␣a␣stupid␣tes
19
This␣is␣a␣stupid␣te
5
This␣
4
This
3
Thi
0
```

```
-> A␣stitch␣in␣time␣saves␣nine.
-> i
12
A␣stitch␣in␣
11
A␣stitch␣in
10
A␣stitch␣i
9
A␣stitch␣
2
A␣
0
```

3 Coding (20 Points)

In this exercise you will be completing the following program by coding the three methods, `IsDartInCircle`, `rand`, and `ThroughDart`. Write your code in the code boxes on the next page.

IsDartInCircle This method is to take in as parameters two doubles, x and y and output a boolean. If the point (x, y) is inside the circle of radius 1 around the origin then the output should be true and if the point is not in that circle the output should be false. Recall from class that (x, y) is inside the circle if $x^2 + y^2 < 1$.

rand This method is to take in as parameters two doubles, l and u and output a double. The method is to create a random number between l and u. This is a decimal random number not an integer. So the call of `rand(-3, 5)` will create a random decimal number between -3 and 5, so an output of 2.983, -1.11224, 4.104, . . . would all be legitimate.

ThrowDart This method will take in a single integer parameter n and return an integer. The method is to throw a random dart at the square $[-1, 1] \times [-1, 1]$ (that is, the point the dart hits (x, y) is random with $-1 < x < 1$ and $-1 < y < 1$) and count the number of times the dart lands inside the circle of radius 1. This number of times that the dart is inside the circle is the number returned by the method. This method must call the other two methods. It should call rand to do the randomizing of x and y and then call the IsDartInCircle method to determine if we have a hit.

```java
import java.util.Scanner;

public class Exam2_Prog1 {

    public static boolean IsDartInCircle(double x, double y) {
        return x * x + y * y < 1;
    }

    public static double rand(double l, double u) {
        return Math.random() * (u - l) + l;
    }

    public static int ThrowDart(int n) {
        int count = 0;
        for (int i = 0; i < n; i++) {
            double x = rand(-1, 1);
            double y = rand(-1, 1);
            if (IsDartInCircle(x, y))
                count++;
        }
        return count;
    }

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

        System.out.print("Number of Darts: ");
        int n = kb.nextInt();
        int hits = ThrowDart(n);
        System.out.println("Darts: " + n);
        System.out.println(" Hits: " + hits);
        System.out.println("Approximation to Pi: " + (double) hits / n * 4);
    }
}
```