PRACTICE FOR QUIZ 4
CMPS 12a - Spring 02
Thomas Raffill

Name: _______________________________________
Student ID: ______________________

This exam is closed book, closed notes, no electronic devices. Show all work.
Partial credit given for partial solutions. Presentation counts! Be legible and
coherent for full credit.

Question 1: ____________ (out of 20)
Question 2: ____________ (out of 20)
Question 3: ____________ (out of 20)
Question 4: ____________ (out of 20)
Question 5: ____________ (out of 20)
Question 6: ____________ (out of 20 Extra Credit)

Total: ___________________ (out of 120)
(Anything above 100 counts for extra credit)
1. (20 points)

```java
public class PowerArray {
    public static void main(String[] args) {
        int numLimit = Integer.parseInt(args[0]);
        int powLimit = Integer.parseInt(args[1]);
        int[][] powerArray = new int[numLimit][powLimit];
        fillPowerArray(powerArray, numLimit, powLimit);
    }

    public static void fillPowerArray(int[][] powArray, int numLim, int powLim) {
        for (int i = 0; i < powLim; i++) {
            for (int j = 0; j < numLim; j++) {
                powArray[i][j] = power(i, j);
                System.out.print(powArray[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static int power(int u, int v) {
        int result = 1;
        for (int i = 1; i <= v; i++)
            result *= u;
        return result;
    }
}
```

This code tries to fill and print out an array of powers of numbers. If you run `java PowerArray 4 4` it will print out:

```
1 0 0 0
1 1 1 1
1 2 4 8
1 3 9 27
```

But if you run `java PowerArray 3 4` you see the following error message:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3
at PowerArray.fillPowerArray(PowerArray.java:12)
at PowerArray.main(PowerArray.java:6)
```

Explain what this Exception means and how you should fix it.
2. (20 points)
You are writing a class with a private int instance variable called idNumber. Now you want to write a “set” method:

public void setID(int idNumber) which sets the value to the given parameter. Write this function.

**HINT:** You need to do something to handle the issue of eclipsing.
3. (20 points)
You are writing a class called Circle with instance variables called radius, circumference, area. You want to enforce that the circumference is always $2\pi$ times the radius and the area is always $\pi$ times the radius squared.

(a) Write a constructor for this class that takes an int parameter for the radius and initializes all three instance variables correctly.

(b) Write a public method setRadius that takes an int parameter for the radius and updates all three instance variables correctly.

**HINT:** Use Math.PI for the value of $\pi$. 
Name:______________________________

4. (20 points)
What does the following piece of code do? **WARNING!** Look carefully at the punctuation!

```java
int i = 1;
boolean keepLooping = true;

while (keepLooping);
{
    if (i < 1000)
    {
        System.out.println(i++);
    }
}
```
5. (20 points)
What does the following piece of code do? **HINT:** What does the keyword `new` do?

```java
public class TestString
{
    public static void main (String[] args)
    {
        String s1 = new String("abc");
        String s2 = new String("abc");
        if (s1 == s2)
        {
            System.out.println("The two strings are the same.");
        }
        else
        {
            System.out.println("The two strings are different.");
        }
    }
}
```
6. **Extra Credit:** (20 points)
   Write a class called PlanePoint to model a point in a plane. It should have private instance variables of type double called xCoordinate and yCoordinate. It should have a constructor that takes two double values and returns a point with the given coordinates. It should have public get and set methods for each coordinate. It should have a public instance method called distanceFrom which takes a PlanePoint parameter and gives the distance between the instance and the given point, and a public static method called distanceBetween which takes two PlanePoints and gives the distance between them. The distance functions should return a double and should use the formula \( d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \) to compute the distance.