Problem 1: Class Rectangle

Write the following class: Rectangle.

The class is to have the following private data members: *length*, which stores the value of the length of the rectangle, *width* which stores the value of the width of the rectangle, and each are of type double. Additionally, the class is to have a static data member named *numberOfRectangles*, of type private.

The class is to have two constructors. The first is to take no arguments, and will initialize the private data members to 0.0. The second constructor is to take two arguments, the first to be used to initialize the data member *width*, the second to initialize the data member *length*. Additionally, each constructor will implement the data member *numberOfRectangles* by 1 to indicate the instantiation of a new object.

The class is to have the following get/set methods:

setWidth( )
//a void method that takes a single argument of type double
//the method is to set the class variable *width* to the value of the argument

setLength( )
//a void method that takes a single argument of type double
//the method is to set the class variable *length* to the value of the argument

getWidth( )
//the method is to take no arguments
//the method is to return the width of the rectangle

getLength( )
//the method is to take no arguments
//the method is to return the length of the rectangle

getArea( )
//the method is to take no arguments
//the method is to return the area of the rectangle

getPerimeter( )
//the method is to take no arguments
//the method is to return the area of the rectangle

getNumberOfRectangles( )
//the method is to take no arguments
//the method is to return the current number of Rectangle objects

The class is to Override the following two Object class methods:

equals( )
//the method is a boolean returning method
//the method takes a single argument, an object of type Rectangle
//the method returns *true* if the values stored in each object are identical
//otherwise, the method returns false

toString( )
//the method is to return the following data

```
The dimensions of the rectangle are as follows:
width = objectWidthValue
length = objectLengthValue
area = objectAreaValue
```

Problem 2: Program Output

Using the Rectangle class written above, please provide the output for the following code.

```
import java.util.Scanner;

public class RectangleApp
{
   public static void main(String [] args)
   {
      //instantiate two Rectangle objects
      Rectangle r1 = new Rectangle();  //no-arg constructor
      Rectangle r2 = new Rectangle(6.0, 3.0); //two-arg constructor
      Scanner kbd = new Scanner(System.in);
      double a = 0.0;
      double b = 0.0;

      System.out.println("Please enter a width and length");
      a = kbd.nextDouble();
      b = kbd.nextDouble();

      Rectangle r7 = new Rectangle(a, b);

      System.out.println("DISPLAY r1 AND r2 DATA\n\n");
      System.out.println("r1 length = " + r1.getLength());
      System.out.println("r1 width = " + r1.getWidth());


      //see data in r1 and r2
      System.out.println("The values stored in r1 are: " +
                  "\nlength = " + r1.getLength() +
                  "\nwidth = " + r1.getWidth());
      r1.printAreaMessage();

      System.out.println("The values stored in r2 are: " +
                  "\nlength = " + r2.getLength() +
                  "\nwidth = " + r2.getWidth());
      r2.printAreaMessage();
      System.out.println("\nThe number of Rectangle objects now is: "
                           + Rectangle.getNumberOfRectangles() );

      //test for equality
      System.out.println("\n\nTEST FOR EQUALITY OF r1 AND r2\n\n");
      if(r1.equals(r2))
         System.out.println("r1 and r2 are equal");
      else
         System.out.println("r1 and r2 are not equal");
      //set new data values for r1
      r1.setLength(2.0);
      r1.setWidth(3.0);

      System.out.println("The values stored in r1 are: " +
                  "\nlength = " + r1.getLength() +
                  "\nwidth = " + r1.getWidth());
      r1.printAreaMessage();

      //assign the values of r2 to r1
      System.out.println("\n\nASSIGN NEW VALUES TO r1 AND r2\n\n");
      r1.setLength(r2.getLength());
      r1.setWidth(r2.getWidth());

      System.out.println("The values stored in r1 are: " +
                  "\nlength = " + r1.getLength() +
                  "\nwidth = " + r1.getWidth());
```

```
      r1.printAreaMessage();

      //demo Object level toString()
      System.out.println("\n\nDISPLAY OBJECT CLASS toString( ) INFORMATION\n\n");
      System.out.println("Object toString method");
      System.out.println(r1.toString());

      //demo Overwritten toString() method
      System.out.println("\n\nDISPLAY RECTANGLE CLASS OVERRIDDEN toString( ) INFORMATION\n\n");
      System.out.println(r1);

      System.out.println("\n\nTEST OBJECT CLASS OVERRIDDEN EQUALS METHOD\n\n");
      if(r1.equals(r2))
         System.out.println("r1 and r2 are equal");
      else
         System.out.println("r1 and r2 are not equal");


     Rectangle r4 = new Rectangle();


     Rectangle [ ] ra = {new Rectangle(), new Rectangle(3, 5), new Rectangle()};

     System.out.println(Rectangle.getNumberOfRectangles());
     System.out.println(r1.getNumberOfRectangles());
     System.out.println(ra[0].getNumberOfRectangles());

   }
}
```

Problem 2: Class Person

Write the following class: Person.

Data Members
The class Person is to have three data members: firstName of type String, lastName of type String and id of type int representing a mock social security number. Each data member has an access specifier of type private.

Constructors
The class is to have two constructors:. The first is a no-arg constructor that initializes each string to an empty string and the integer data member to zero. The second constructor takes three arguments, each representing one of the class data members. The arguments are to be listed in the order of firstName, lastName, id.

Get/Set Method
Write appropriate get/set methods for each data member

Auxiliary Methods
Override the Object toString( ) method to display the following information:

First Name: *display firstName data*
Last Name: *display lastName data*
ID: *display id data*

Override the Object equals( ) method so that the method returns true if each data member of the calling object and the corresponding data members of the argument object are identical.