Class Shape


```
public class Shape
{
    private int xCoord;
    private int yCoord;
    protected String color="black";
    protected String id="";

    Shape(int x, int y, String c)
    {
        xCoord=x;
        yCoord=y;
        color=c;
    }

    Shape()
    {
        xCoord=0;
        yCoord=0;
    }

    public void setColor(String newColor)
    {
        color=newColor;
    }

    public String getColor()
    {
        return color;
    }

    public void setId(String newId)
    {
        id=newId;
    }

    public String getId()
    {
        return id;
    }

    public void setX(int newX)
    {
        xCoord=newX;
    }

    public void setY(int newY)
    {
        yCoord=newY;
    }

    public int getX()
    {
        return xCoord;
```

```
    }

    public int getY()
    {
        return yCoord;
    }

    public void moveTo(int newX, int newY)
    {
        xCoord=newX;
        yCoord=newY;
    }

    public void print()
    {
        System.out.println("Coordinates (x,y) = (" + xCoord + "," + yCoor
d +")");
        System.out.println("Color: " + color);
    }

    public double area()
    {
        return 0;
    }

    public double perimeter()
    {
        return 0;
    }

}
```

Class Rectangle

```
public class Rectangle extends Shape
{
    private int width;
    private int length;
    private static int rectangleCount;

    Rectangle(int len, int wid)
    {
        super(0,0,"red");
        length=len;
        width=wid;
        rectangleCount++;
        setId("Rectangle-"+rectangleCount);
    }

    Rectangle()
    {
        // Special case of the above constructor with 'length=1' & 'width
=1'
        this(1, 1);
    }

    public double area()
    {
        return (double) width*length;
    }

        public double perimeter()
    {
        return (double) 2*(length+width);
    }

    public void print()
    {
        System.out.println("\n---Rectange----");
        System.out.println("Id: " + id);
        System.out.println("Length: " + length);
        System.out.println("Width: " + width);
        super.print();
        System.out.println("Area: " + area());
        System.out.println("Perimeter: " + perimeter());
    }
}
```

Class Circle


```
public class Circle extends Shape
{
    private int radius;
    private static int circleCount;
    private final double PI=3.14;

    Circle(int rad)
    {
        super(0,0,"red");
        radius=rad;
        circleCount++;
        setId("Circle-"+circleCount);
    }

    Circle()
    {
        // Special case: previous constructor with 'radius=1'
        this(1);
    }

    public double area()
    {
        return (double) PI*radius*radius;
    }

    public double perimeter()
    {
        return (double) 2*PI*radius;
    }

    public void print()
    {
        System.out.println("\n---Circle----");
        System.out.println("Id: " + id);
        System.out.println("Radius: " + radius);
        super.print();
        System.out.println("Area: " + area());
        System.out.println("Perimeter: " + perimeter());
    }
}
```

Class DatabaseOfShapes

```
import java.util.ArrayList;

public class DatabaseOfShapes
{
    private ArrayList db;

    DatabaseOfShapes()
    {
        db=new ArrayList();
    }

    public void add(Shape s)
    {
        db.add(s);
    }

    public void listAll()
    {
        System.out.println("\n===List of Shapes===");
        for (int i=0; i<db.size(); i++)
            ((Shape)db.get(i)).print();
    }

    public void printStatistics()
    {
        double totalArea=0;
        double totalPerimeter=0;

        System.out.println("\n===Database Statistics===");
        System.out.println("Number of shapes: " + db.size());

        for (int i=0; i<db.size(); i++)
            totalArea+= ((Shape)db.get(i)).area();
        System.out.println("Total Area: " + totalArea);

        for (int i=0; i<db.size(); i++)
            totalPerimeter+= ((Shape)db.get(i)).perimeter();
        System.out.println("Total perimeter: " + totalPerimeter);
    }

}//Database
```