```java
import java.sql.SQLOutput;

public class Shape {
    private String color;
    protected double area = 1.0;
    protected double base = 1.0;
    protected double width = 1.0;
    protected double height = 1.0;

    public Shape(String color){
        this.color=color;
    }
    public Shape(){}
    public Shape(String color, double area, double base, double width, double height){
        this.color=color;
        this.area=area;
        this.base=base;
        this.width=width;
        this.height=height;
    }

    public void setColor(String color){
        this.color=color;
    }
    public void setArea(double area){
        this.area = area;
    }
    public void setBase(double base){
        this.base = base;
    }
    public void setWidth(double width){
        this.width=width;
    }
    public void setHeight(double height){
        this.height=height;
    }
    @Override
    public String toString(){
        return "Shape[color=" + color+"]";
    }
```

```
41      public double getArea(){
42          System.out.println("Shape unknown! Cannot
    compute area!");
43          return 0;
44      }
45      public void displayshapeName(){
46          System.out.println("I am a Shape.");
47      }
48  }
49
```

```java
public class Circle extends Shape{
    protected double radius;
    private final double PI = Math.PI;

    public Circle(double radius){
        this.radius=radius;
    }

    public Circle(double radius, double height){
        this.radius=radius;
        super.height = height;
    }

//    @Override
    public double getArea() {
        super.area = PI*Math.pow(this.radius, 2);
        return super.area;
    }

    @Override
    public void displayshapeName(){
        System.out.println("Drawing a Circle of radius " + this.radius);
    }

    @Override
    public String toString(){
        return "Circle[ radius = " + radius + super.toString() + "]";
    }
}

```

```java
public class Cylinder extends Circle{
    private final double PI = Math.PI;
    public Cylinder(double radius, double height){
        super(radius,height);
    }
    public Cylinder(double radius){
        super(radius);
    }

    public double getVolume(){
        return PI*Math.pow(super.radius,2) * super.height;
    }

    public double getSurfaceArea(){
        return 2.0 * Math.PI*super.radius*super.height;
    }

    @Override
    public void displayshapeName() {
        System.out.println("Drawing a Cylinder for radius " + super.radius);
    }
    public String toString(){
        return "radius is: " + super.radius + " height is : " + super.height;
    }
}

```

```java
public class myRunner {
    public static void main(String[] args){
        Circle c = new Circle(100);
        System.out.println("Area of Circle " + c.
getArea());

        Shape sObj = new Shape();
        sObj.displayshapeName();
        System.out.println(sObj instanceof Shape);

        System.out.println("++++++++++++++++++");

        Shape shapeCircleObj = new Circle(100); //
upCasting
        shapeCircleObj.displayshapeName();
        System.out.println("Area of Circle " +
shapeCircleObj.getArea());
        System.out.println(shapeCircleObj); // Run
circle's toString()
        // Use instanceof operator for validation
        System.out.println(shapeCircleObj
instanceof Circle); //true
        System.out.println("----------------");
        Shape shapeRectangleObj = new Rectangle("
Red");
        shapeRectangleObj.displayshapeName();
        shapeRectangleObj.setHeight(2);
        shapeRectangleObj.setWidth(2);
        System.out.println("Area of Rectangle is "
 + shapeRectangleObj.getArea());
        System.out.println(shapeRectangleObj); //
Run Rectangle's toString()
        // Use instanceof operator for Validation
        System.out.println(shapeRectangleObj
instanceof Rectangle);
        System.out.println(sObj instanceof
Rectangle);

        System.out.println("__________________");
        Shape shapeTriangleObj = new Triangle("Blue
"); //UpCasting
```

```
31          shapeTriangleObj.displayshapeName();
32          shapeTriangleObj.setHeight(2);
33          shapeTriangleObj.setBase(3);
34          System.out.println("Area of Triangle is "
    + shapeTriangleObj.getArea());
35          System.out.println(shapeTriangleObj);
36
37          System.out.println(shapeTriangleObj
   instanceof Triangle);
38          System.out.println(sObj instanceof Triangle
   );
39
40          System.out.println("---------------");
41
42          Cylinder cylinderShape = new Cylinder(3);
   //upcasting
43          cylinderShape.displayshapeName();
44          cylinderShape.setHeight(3);
45          System.out.println("Area of cylinder is "
    + cylinderShape.getVolume());
46          System.out.println(cylinderShape);
47      }
48 }
49
```

```java
public class
Triangle extends Shape {
    public Triangle(){

    }
    public Triangle(String color, double area,
  double base, double width, double height){
        super(color, area, base, width, height);
    }
    public Triangle(String color){
        super(color);
    }

    @Override
    public void setBase(double base) {
        super.setBase(base);
    }

    @Override
    public void setWidth(double width) {
        super.setWidth(width);
    }

    @Override
    public double getArea() {
        return 0.5*base*height;
    }

    @Override
    public void displayshapeName() {
        System.out.println("I am a triAngle");
    }

    @Override
    public String toString() {
        return "Triangle[base=" + base + ",height="
   + height + "," + super.toString() + "]";
    }
}

```

```java
public class Rectangle extends Shape {
    public Rectangle(String color){
        super(color);
    }

    public Rectangle(){

    }
    public Rectangle(String color, double area, double base, double width, double height){
        super(color, area, base, width, height);
    }

    @Override
    public void setBase(double base){
        super.base = base;
    }

    @Override
    public void setWidth(double width) {
        super.setWidth(width);
    }

    @Override
    public double getArea() {
        return super.getArea();
    }
    public double perimeter(){
        super.area = super.width *super.height;
        return super.area;
    }

    @Override
    public String toString() {
        return "Rectangle[height=" +height +",width =" +width +"," + super.toString() +"]";
    }
}

```