

```
1 import java.sql.SQLOutput;  
2  
3 public class Shape {  
4     private String color;  
5     protected double area = 1.0;  
6     protected double base = 1.0;  
7     protected double width = 1.0;  
8     protected double height = 1.0;  
9  
10    public Shape(String color){  
11        this.color=color;  
12    }  
13    public Shape(){}  
14    public Shape(String color, double area, double  
base, double width, double height){  
15        this.color=color;  
16        this.area=area;  
17        this.base=base;  
18        this.width=width;  
19        this.height=height;  
20    }  
21  
22    public void setColor(String color){  
23        this.color=color;  
24    }  
25    public void setArea(double area){  
26        this.area = area;  
27    }  
28    public void setBase(double base){  
29        this.base = base;  
30    }  
31    public void setWidth(double width){  
32        this.width=width;  
33    }  
34    public void setHeight(double height){  
35        this.height=height;  
36    }  
37    @Override  
38    public String toString(){  
39        return "Shape[color=" + color+"]";  
40    }
```

```
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
```

```
1 public class Circle extends Shape{
2     protected double radius;
3     private final double PI = Math.PI;
4
5     public Circle(double radius){
6         this.radius=radius;
7     }
8
9     public Circle(double radius, double height){
10         this.radius=radius;
11         super.height = height;
12     }
13
14 //    @Override
15     public double getArea() {
16         super.area = PI*Math.pow(this.radius, 2);
17         return super.area;
18     }
19
20     @Override
21     public void displayshapeName(){
22         System.out.println("Drawing a Circle of
radius " + this.radius);
23     }
24
25     @Override
26     public String toString(){
27         return "Circle[ radius = " + radius + super
.toString() + "];"
28     }
29 }
30
```

```
1 public class Cylinder extends Circle{
2     private final double PI = Math.PI;
3     public Cylinder(double radius, double height){
4         super(radius,height);
5     }
6     public Cylinder(double radius){
7         super(radius);
8     }
9
10    public double getVolume(){
11        return PI*Math.pow(super.radius,2) * super.
height;
12    }
13
14    public double getSurfaceArea(){
15        return 2.0 * Math.PI*super.radius*super.
height;
16    }
17
18    @Override
19    public void displayshapeName() {
20        System.out.println("Drawing a Cylinder for
radius " + super.radius);
21    }
22    public String toString(){
23        return "radius is: " + super.radius + "
height is : " + super.height;
24    }
25 }
26
```

```

1 public class myRunner {
2     public static void main(String[] args){
3         Circle c = new Circle(100);
4         System.out.println("Area of Circle " + c.
        getArea());
5
6         Shape sObj = new Shape();
7         sObj.displayshapeName();
8         System.out.println(sObj instanceof Shape);
9
10        System.out.println("++++++++++++++++");
11
12        Shape shapeCircleObj = new Circle(100); //
        upCasting
13        shapeCircleObj.displayshapeName();
14        System.out.println("Area of Circle " +
        shapeCircleObj.getArea());
15        System.out.println(shapeCircleObj); // Run
        circle's toString()
16        // Use instanceof operator for validation
17        System.out.println(shapeCircleObj
        instanceof Circle); //true
18        System.out.println("-----");
19        Shape shapeRectangleObj = new Rectangle("
        Red");
20        shapeRectangleObj.displayshapeName();
21        shapeRectangleObj.setHeight(2);
22        shapeRectangleObj.setWidth(2);
23        System.out.println("Area of Rectangle is "
        + shapeRectangleObj.getArea());
24        System.out.println(shapeRectangleObj); //
        Run Rectangle's toString()
25        // Use instanceof operator for Validation
26        System.out.println(shapeRectangleObj
        instanceof Rectangle);
27        System.out.println(sObj instanceof
        Rectangle);
28
29        System.out.println("-----");
30        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
```

```
1 public class
2 Triangle extends Shape {
3     public Triangle(){
4
5     }
6     public Triangle(String color, double area,
7 double base, double width, double height){
8         super(color, area, base, width, height);
9     }
10    public Triangle(String color){
11        super(color);
12    }
13    @Override
14    public void setBase(double base) {
15        super.setBase(base);
16    }
17
18    @Override
19    public void setWidth(double width) {
20        super.setWidth(width);
21    }
22
23    @Override
24    public double getArea() {
25        return 0.5*base*height;
26    }
27
28    @Override
29    public void displayshapeName() {
30        System.out.println("I am a triAngle");
31    }
32
33    @Override
34    public String toString() {
35        return "Triangle[base=" + base + ",height="
36        + height + ", " + super.toString() + "];"
37    }
38 }
```

```
1 public class Rectangle extends Shape {
2     public Rectangle(String color){
3         super(color);
4     }
5
6     public Rectangle(){
7
8     }
9     public Rectangle(String color, double area,
10    double base, double width, double height){
11         super(color, area, base, width, height);
12     }
13
14     @Override
15     public void setBase(double base){
16         super.base = base;
17     }
18
19     @Override
20     public void setWidth(double width) {
21         super.setWidth(width);
22     }
23
24     @Override
25     public double getArea() {
26         return super.getArea();
27     }
28     public double perimeter(){
29         super.area = super.width *super.height;
30         return super.area;
31     }
32
33     @Override
34     public String toString() {
35         return "Rectangle[height=" +height +",width
36         =" +width +", " + super.toString() +"]";
37     }
38 }
```