

```
1 //TIP To <b>Run</b> code, press <shortcut actionId
  ="Run"/> or
2 // click the <icon src="AllIcons.Actions.Execute
  "/> icon in the gutter.
3 public class Main {
4     public static void main(String[] args) {
5         //TIP Press <shortcut actionId="
  ShowIntentionActions"/> with your caret at the
  highlighted text
6         // to see how IntelliJ IDEA suggests fixing
  it.
7         System.out.printf("Hello and welcome!");
8
9         for (int i = 1; i <= 5; i++) {
10            //TIP Press <shortcut actionId="Debug
  "/> to start debugging your code. We have set one <
  icon src="AllIcons.Debugger.Db_set_breakpoint"/>
  breakpoint
11            // for you, but you can always add more
  by pressing <shortcut actionId="
  ToggleLineBreakpoint"/>.
12            System.out.println("i = " + i);
13        }
14    }
15 }
```

```
1 public abstract class Shape {
2     protected String color;
3     protected double height;
4     protected double width;
5     protected double base;
6
7     public void setColor(String color){
8         this.color=color;
9     }
10    public void setHeight(double height){
11        this.height=height;
12    }
13    public void setWidth(double width){
14        this.width=width;
15    }
16    public void setBase(double base){
17        this.base=base;
18    }
19    public abstract double getArea();
20    public String toString(){
21        return "Shape[color=" +color+"]";
22    }
23    public void displayShapeName(){
24        System.out.println("I am a Shape.");
25    }
26 }
27
```

```
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    public double getArea(){
14        double area = PI*Math.pow(this.radius,2);
15        return area;
16    }
17
18    @Override
19    public void displayShapeName(){
20        System.out.println("Drawing a Circle of
radius " + this.radius);
21    }
22
23    @Override
24    public String toString(){
25        return "Circle[ radius=" +radius + super.
toString()+ "];"
26    }
27 }
28
```

```
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.
5             getArea());
6         System.out.println("+++++++");
7         Shape shapeCircleObj = new Circle(100);
8         shapeCircleObj.displayShapeName();
9         System.out.println("Area of Circle " +
10             shapeCircleObj.getArea());
11         System.out.println(shapeCircleObj
12             instanceof Circle);
13         System.out.println("+++++");
14         Shape shapeRectangleObj = new Rectangle("
15             Red");
16         shapeRectangleObj.displayShapeName();
17         shapeRectangleObj.setHeight(2);
18         shapeRectangleObj.setWidth(4);
19         System.out.println("Area of Rectangle is "
20             + shapeRectangleObj.getArea());
21         System.out.println(shapeCircleObj
22             instanceof Rectangle);
23         System.out.println("-----");
24         Shape shapeTriangleObj = new Triangle("Blue
25             ");
26         shapeTriangleObj.displayShapeName();
27         shapeTriangleObj.setHeight(10);
28         shapeTriangleObj.setBase(15);
29         System.out.println("Area of Triangle is " +
30             shapeTriangleObj.getArea());
31         System.out.println(shapeTriangleObj);
32     }
33 }
```

```
1 public class Triangle extends Shape{
2     public Triangle(){}
3     public Triangle(String color){
4         super.color =color;
5     }
6     public void setBase(int base){
7         this.base=base;
8     }
9     @Override
10    public double getArea(){
11        return .5*super.base *super.height;
12    }
13    @Override
14    public void displayShapeName(){
15        System.out.println("I am a triangle");
16    }
17
18    @Override
19    public String toString() {
20        return "Triangle[base="+super.base+",height
21        =" +super.height +", " +super.toString() + "];
22    }
23 }
```

```
1 public class Rectangle extends Shape{
2     public Rectangle(String color){
3         super.color=color;
4
5     }
6     public Rectangle(){
7
8     }
9     public Rectangle(String color, double width,
double height){
10         super.height = height;
11         super.width =width;
12         super.color =color;
13     }
14     @Override
15     public double getArea(){
16         return super.width *super.height;
17     }
18     @Override
19     public void displayShapeName(){
20         System.out.println("I am a Rectangle");
21     }
22
23     @Override
24     public String toString(){
25         return "Rectangle[height=" +height+",width
=" +width+", " + super.toString()+""];
26     }
27 }
28
```