

Factory pattern with shape factory area

Step-1: Shape Interface

```
public interface Shape {  
    double getArea();  
}
```

- ◆ সব shape-এর জন্য common interface
 - ◆ Factory শৰ্ক্ষণে `Shape` return করবে → loose coupling
-

Step-2: Circle Class

```
public class Circle implements Shape {  
  
    private double radius;  
  
    // default constructor  
    public Circle() {  
        this.radius = 1.0;    // default value  
    }  
  
    @Override  
    public double getArea() {  
        return Math.PI * radius * radius;  
    }  
}
```

 Formula: πr^2

 radius = 1 (default)

Step-3: Rectangle Class

```
public class Rectangle implements Shape {  
  
    private double width;  
    private double height;  
  
    public Rectangle() {  
        this.width = 2.0;      // default  
        this.height = 1.5;     // default  
    }  
  
    @Override  
    public double getArea() {  
        return width * height;  
    }  
}
```

📌 Formula: width × height

🌐 Step-4: Triangle Class

```
public class Triangle implements Shape {  
  
    private double a, b, c;  
  
    public Triangle() {  
        this.a = 3.0;  
        this.b = 4.0;  
        this.c = 5.0;    // valid triangle  
    }  
  
    @Override  
    public double getArea() {  
        double s = (a + b + c) / 2;  
        return Math.sqrt(s * (s - a) * (s - b) * (s - c));  
    }  
}
```

📌 Heron's Formula

📌 $s = (a + b + c) / 2$

💻 Step-5: ShapeFactory Class (Core of Factory Pattern)

```
public class ShapeFactory {  
  
    public Shape getShape(String shapeType) {  
  
        if (shapeType == null)  
            return null;  
  
        if (shapeType.equalsIgnoreCase("CIRCLE")) {  
            return new Circle();  
        }  
        else if (shapeType.equalsIgnoreCase("RECTANGLE")) {  
            return new Rectangle();  
        }  
        else if (shapeType.equalsIgnoreCase("TRIANGLE")) {  
            return new Triangle();  
        }  
  
        return null;  
    }  
}
```

🔑 এখানে কী হচ্ছে:

- new keyword client জানে না

- object creation centralized
 - client শৈক্ষণিক shape name দেয়
-

▶ Step-6: Demo / Main Class

```
public class FactoryPatternDemo {  
  
    public static void main(String[] args) {  
  
        ShapeFactory factory = new ShapeFactory();  
  
        Shape circle = factory.getShape("CIRCLE");  
        System.out.println("Circle Area = " + circle.getArea());  
  
        Shape rectangle = factory.getShape("RECTANGLE");  
        System.out.println("Rectangle Area = " + rectangle.getArea());  
  
        Shape triangle = factory.getShape("TRIANGLE");  
        System.out.println("Triangle Area = " + triangle.getArea());  
    }  
}
```

💻 Output (Approximate)

```
Circle Area = 3.141592653589793  
Rectangle Area = 3.0  
Triangle Area = 6.0
```

