```
┌─────────────────────┐
│   Factory Pattern   │
│       Demo          │
├─────────────────────┤
│                     │
├─────────────────────┤
│   +main(): void     │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│   Shape Factory     │
├─────────────────────┤
│                     │
├─────────────────────┤
│  +getShape():Shape  │
└─────────────────────┘
```

<<interface>>
Shape

+ draw() : void

Implements

Implements

Implements

Circle
+ draw() : void

Square
+ draw() : void

Rectangle
+ draw() : void

```java
Circle.java U ×

Circle.java > Circle
1   public class Circle implements Shape {
2       @Override
3       public void draw() {
4           System.out.println(x: "Inside Circle::draw() method.");
5       }
6   }
```

```java
FactoryPatternDemo.java U ×

FactoryPatternDemo.java > FactoryPatternDemo
1    public class FactoryPatternDemo {
         Run | Debug
2        public static void main(String[] args) {
3            ShapeFactory shapeFactory = new ShapeFactory();
4            Shape shape1 = shapeFactory.getShape(shapeType: "CIRCLE");
5            shape1.draw();
6            Shape shape2 = shapeFactory.getShape(shapeType: "RECTANGLE");
7            shape2.draw();
8            Shape shape3 = shapeFactory.getShape(shapeType: "SQUARE");
9            shape3.draw();
10       }
11   }
```

Rectangle.java U ✕

Rectangle.java > Rectangle

```java
public class Rectangle implements Shape {
    @Override
    public void draw() {
        System.out.println(x: "Inside Rectangle::draw() method.");
    }
}
```

Shape.java U ✕

Shape.java > Shape

```java
public interface Shape {
    void draw();
}
```

```java
public class ShapeFactory {
    public Shape getShape(String shapeType) {
        if (shapeType == null) {
            return null;
        }
        if (shapeType.equalsIgnoreCase(anotherString: "CIRCLE")) {
            return new Circle();
        } else if (shapeType.equalsIgnoreCase(anotherString: "RECTANGLE")) {
            return new Rectangle();
        } else if (shapeType.equalsIgnoreCase(anotherString: "SQUARE")) {
            return new Square();
        }
        return null;
    }
}
```

```java
public class Square implements Shape {
    @Override
    public void draw() {
        System.out.println(x: "Inside Square::draw() method.");
    }
}
```

```
cd "/Users/tententgc/Documents/GitHub/comsci-path/css222/uml/umlcode/" && javac FactoryPatternDemo.java && java FactoryP
emo
> cd "/Users/tententgc/Documents/GitHub/comsci-path/css222/uml/umlcode/" && javac FactoryPatternDemo.java && java Factor
nDemo
Inside Circle::draw() method.
Inside Rectangle::draw() method.
Inside Square::draw() method.
   ~/Doc/Gi/comsci-path/css222/u/umlcode   on   main !9 ?16                              base    at 12:52:
```