Task 2: Factory Method

Create a ShapeFactory class that encapsulates the object creation logic of different Shape objects like Circle, Square, and Rectangle.

```
public interface Shape {
  void draw();
}
public class Circle implements Shape {
  @Override
  public void draw() {
    System.out.println("Inside Circle::draw() method.");   }
}
public class Square implements Shape {
  @Override
  public void draw() {
    System.out.println("Inside Square::draw() method.");   }
}
public class Rectangle implements Shape {
  @Override
  public void draw() {
    System.out.println("Inside Rectangle::draw() method.");
  }
}
public class ShapeFactory {
  public Shape getShape(String shapeType) {
    if (shapeType == null) {
      return null;
    } else if (shapeType.equalsIgnoreCase("CIRCLE")) {
      return new Circle();
    } else if (shapeType.equalsIgnoreCase("SQUARE")) {
      return new Square();
    } else if (shapeType.equalsIgnoreCase("RECTANGLE")) {
            return new Rectangle();
        }
        return null;
    }
}
public class Main {
    public static void main(String[] args) {
        ShapeFactory shapeFactory = new ShapeFactory();
        Shape circle = shapeFactory.getShape("CIRCLE");
        if (circle != null) {
            circle.draw();
        }
        Shape square = shapeFactory.getShape("SQUARE");
        if (square != null) {
            square.draw();
        }
        Shape rectangle = shapeFactory.getShape("RECTANGLE");
        if (rectangle != null) {
            rectangle.draw();
        }
    }
}
```