

DAY 28:

ASSIGNMENT 2:

Task 2: Factory Method

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

ANSWER:

```
// Shape.java
```

```
abstract class Shape {  
    public abstract void draw();  
}
```

```
// Circle.java
```

```
class Circle extends Shape {  
    @Override  
    public void draw() {  
        System.out.println("Drawing a Circle");  
    }  
}
```

```
// Square.java
```

```
class Square extends Shape {  
    @Override  
    public void draw() {  
        System.out.println("Drawing a Square");  
    }  
}
```

```
// Rectangle.java
```

```
class Rectangle extends Shape {  
    @Override
```

```
public void draw() {  
    System.out.println("Drawing a Rectangle");  
}  
}
```

// ShapeFactory.java

```
class ShapeFactory {  
    // Factory method to create shapes  
    public Shape createShape(String shapeType) {  
        if (shapeType == null) {  
            return null;  
        }  
        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;  
    }  
}
```

// Main.java

```
public class Main {  
    public static void main(String[] args) {  
        ShapeFactory shapeFactory = new ShapeFactory();  
  
        // Create a Circle  
        Shape shape1 = shapeFactory.createShape("CIRCLE");  
        shape1.draw(); // Output: Drawing a Circle
```

```
// Create a Square  
Shape shape2 = shapeFactory.createShape("SQUARE");  
shape2.draw(); // Output: Drawing a Square  
  
// Create a Rectangle  
Shape shape3 = shapeFactory.createShape("RECTANGLE");  
shape3.draw(); // Output: Drawing a Rectangle  
}  
}
```