Factory pattern is one of the most used design patterns in Java. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.

In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface.



interface Shape{

void draw();

}

class Rectangle implements Shape{

@Override

public void draw() {

System.out.println("Inside rectangle draw method");

}

}

class Square implements Shape{

@Override

public void draw() {

System.out.println("Inside Square draw method");

}

}

class Circle implements Shape{

@Override

public void draw() {

System.out.println("Inside Circle draw method");

}

}

class ShapeFactory{

public Shape getshape(String shape) {

switch(shape) {

case "RECTANGLE" : return new Rectangle();

case "CIRCLE" : return new Circle();

case "SQUARE" : return new Square();

}

return null;

}

}

public class FactoryPatternDemo {

public static void main(String[] args) {

ShapeFactory sf = new ShapeFactory();

Shape s = sf.getshape("CIRCLE");

s.draw();

}

}