**package** designPattern3;

**interface** Shape {

**void** draw();

}

**package** designPattern3;

**public** **class** Rectangle **implements** Shape{

@Override

**public** **void** draw() {

System.***out***.println("Inside Rectangle");

}

}

**package** designPattern3;

**public** **class** Square **implements** Shape{

@Override

**public** **void** draw() {

System.***out***.println("Inside Square");

}

}

**package** designPattern3;

**public** **class** RoundedRectangle **implements** Shape{

@Override

**public** **void** draw() {

System.***out***.println("Inside Rounded Rectangle");

}

}

**package** designPattern3;

**public** **class** RoundedSquare **implements** Shape{

@Override

**public** **void** draw() {

System.***out***.println("Inside Rounded Square");

}

}

**package** designPattern3;

**public** **abstract** **class** AbstractFactory {

**abstract** Shape getShape(String ShapeType);

}

**package** designPattern3;

**public** **class** ShapeFactory **extends** AbstractFactory {

@Override

Shape getShape(String ShapeType) {

**if**(ShapeType==**null**)

{

**return** **null**;

}

**if**(ShapeType.equalsIgnoreCase("Rectangle"))

{

**return** **new** Rectangle();

}

**if**(ShapeType.equalsIgnoreCase("Square"))

{

**return** **new** Square();

}

**return** **null**;

}

}

**package** designPattern3;

**public** **class** RoundedShapeFactory **extends** AbstractFactory{

@Override

Shape getShape(String ShapeType) {

**if**(ShapeType.equalsIgnoreCase("Rounded Rectangle"))

{

**return** **new** RoundedRectangle();

}

**if**(ShapeType.equalsIgnoreCase("Rounded Square"))

{

**return** **new** RoundedSquare();

}

**return** **null**;

}

}

**package** designPattern3;

**public** **class** AbstractFactoryDemo {

**public** **static** **void** main(String[] args) {

AbstractFactory shapeFactory=FactoryProducer.*getFactory*(**false**);

Shape shape1=shapeFactory.getShape("Rectangle");

shape1.draw();

Shape shape2=shapeFactory.getShape("Square");

shape2.draw();

AbstractFactory shapeFactory1=FactoryProducer.*getFactory*(**true**);

Shape shape3=shapeFactory1.getShape("Rounded Rectangle");

shape3.draw();

Shape shape4=shapeFactory1.getShape("Rounded Square");

shape4.draw();

}

}

**package** designPattern3;

**public** **class** FactoryProducer {

**public** **static** AbstractFactory getFactory(**boolean** rounded)

{

**if**(rounded)

{

**return** **new** RoundedShapeFactory();

}

**else**

{

**return** **new** ShapeFactory();

}

}

}