

JAVA means DURGA SOFT

Design Patterns

Core Level Design Patterns

4. Factory Pattern



India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

Factory Pattern

Sometimes, an Application (or framework) at runtime, cannot anticipate the class of object that it must create. The Application (or framework) may know that it has to instantiate classes, but it may only know about abstract classes (or interfaces), which it cannot instantiate. Thus the Application class may only know **when** it has to instantiate a new Object of a class, not **what kind of** subclass to create. Hence a class may want it's subclasses to specify the objects to be created.

The factory pattern design pattern handles these problems by defining a separate method for creating the objects, these methods optionally accept parameters defining how the object is created, and then return the created object.

Problem: Creating multiple objects for multiple subclasses and utilizing one of them based on the supplied data is wrong methodology because the remaining objects became unnecessarily created objects.

Solution: Use Factory Pattern, here the method of super class or some other class instantiates/creates one of the several sub classes object based on the supplied data provide by user (at compile time/at runtime) that means objects for remaining sub classes will not be created.

Def: Define an interface for creating an object, but let the subclasses decide which class to instantiate. The Factory method lets a class defer instantiation to subclasses.

A Simple Factory pattern returns an instance of one of several possible (sub) classes depending on the data provided to it. Usually all classes that it returns have a common parent class and common methods, but each performs a task differently and is optimized for different kinds of data.

Intent:

- ✓ Creates objects without exposing the instantiation logic to the client.

- ✓ Refers to the newly created object through a common interface.

Java Real Time Tools



JAVA TOOLS Means DURGASOFT

Multiple Faculty Members only For JAVA TOOLS

Online Training Class Room Training

DURGA SOFTWARE SOLUTIONS

www.durgasoftonlinetraining.com durgasoftonlinetraining@gmail.com Ph: +91- 8885252627 +91- 7207212428

Application Areas:

- A class can't anticipate the class of objects it must create.
- A class specifies its sub-classes to specify which objects to create.
- In programmer's language, you can use factory pattern where you have to create an object of any one of sub-classes depending on the data provided.

Ex:

In JDBC applications, **DriverManager.getConnection()** method logic is nothing but factory pattern logic because based on the argument values that are supplied, it creates and returns one of the JDBC driver class that implements the `java.sql.Connection` interface.

- I. `Connection con=Drivermanager.getConnection("jdbc:odbc:durga","uname","pwd");`
 - Here `getConnection(____)` call returns the object of Type-1 JDBC driver class i.e. **`sun.jdbc.odbc.JdbcOdbcConnection`** that implements `java.sql.Connection` interface.
- II. `Connection con= DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "uname","pwd");`

- Here getConnection(____) call returns the object of Type-4 JDBC driver class i.e. **oracle.jdbc.driver.OracleConnection** that implements java.sql.Connection interface.

Sample Code:

```
// FactoryPatternTest.java
```

```
abstract class ShapeEx
```

```
{
```

```
    public abstract void draw();
```

```
};
```

```
class Line extends ShapeEx
```

```
{
```

```
    public Line()
```

```
    {
```

```
        System.out.println("You are Chosen Line To Draw");
```

```
    }
```

```
    public void draw()
```

```
    {
```

```
        System.out.println("Line: draw()");
```

```
    }
```

```
};
```

```
class Square extends ShapeEx
```

```
{
```

```
    public Square()
```

```
    {
```

```
        System.out.println("You are Chosen Square To Draw");
```

```
    }
```

```
    public void draw()
```

```
    {
```

```
        System.out.println("Square: draw()");
```

```
    }  
};  
public class FactoryPatternTest  
{  
    public static void main(String[] args)  
    {  
        FactoryPatternTest fp=new FactoryPatternTest();  
        System.out.println();  
        ShapeEx s=fp.getShape(args[0]); // data is supplied at runtime  
        System.out.println();  
        System.out.println("Object is created for: "+s.getClass());  
    }  
    //Factory Pattern Logic  
    public ShapeEx getShape(String shape)  
    {  
        if(shape.equals("L"))  
            return new Line();  
        else if(shape.equals("S"))  
            return new Square();  
        else  
            return null;  
    }  
}
```

Output:

JAVA Means DURGA SOFT

```
D:\Java\Design Patterns\Programs\Factory Pattern>javac FactoryPatternTest.java
D:\Java\Design Patterns\Programs\Factory Pattern>java FactoryPatternTest L
You are Chosen Line To Draw
Object is created for: class Line
D:\Java\Design Patterns\Programs\Factory Pattern>java FactoryPatternTest S
You are Chosen Square To Draw
Object is created for: class Square
```

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**



USA Ph : 4433326786

E-mail : durgasoftonlinetraining@gmail.com

www.durgajobs.com

Continuous Job Updates for every hour

Fresher Jobs

Govt Jobs

Bank Jobs

Walk-ins

Placement Papers

IT Jobs

Interview Experiences

Complete Job information across India

LEARN FROM EXPERTS ...

COMPLETE JAVA

CORE JAVA, ADV. JAVA, ORACLE, STRUTS, HIBERNATE, SPRING, WEB SERVICES,...

COMPLETE .NET

C#.NET, ASP.NET, SQL SERVER, MVC 5 & WCF

TESTING TOOLS

MANUAL + SELENIUM

ORACLE | D2K

MSBI | SHARE POINT

HADOOP | ANDROID

C, C++, DS, UNIX

CRT & APTITUDE TRAINING

AN ISO 9001:2008 CERTIFIED

DURGA
Software Solutions®

202, 2nd Floor, HUDA Maitrivanam,
Ameerpet, Hyd. Ph: 040-64512786,

9246212143, 8096969696

www.durgasoft.com