

**JAVA means DURGA SOFT**

# Design Patterns

## Core Level Design Patterns

### 6. Builder Design Pattern



**India's No.1 Software Training Institute**

**DURGASOFT**

**www.durgasoft.com Ph: 9246212143 ,8096969696**

## **Builder Design Pattern**

Builder, as the name suggests builds complex objects from simple ones step-by-step. It separates the construction of complex objects from their representation.

**Problem:** In order to complete a task working with simple and individual objects is always complex process to perform.

**Solution:** Use Builder design pattern. Which says creates complex object from simple objects and work with that complex object to complete the task.

As a programmer (user) who utilizes this complex object need not bother about the simple and individual objects that are involved in complex object creation.

**Ex:** When we create object for our frame class (which extends predefined java.awt.Frame class) lots of individual components like Text Field, Button, Label, List Box etc. objects will be created internally. But as a user of this Frame, the Window class object never bother about those individual objects. This is nothing but **Builder Design Pattern**.

**www.durgasoftonlinelearning.com**



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

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

 **USA Ph : 4433326786**

**E-mail : durgasoftonlinelearning@gmail.com**

Builder pattern is used to construct a complex object step by step and the final step will return the object. The process of constructing an object should be generic so that it can be used to create different representations of the same object.

For example, you can consider construction of a home. Home is the final end product (object) that is to be returned as the output of the construction process. It will have many steps, like basement construction, wall construction and so on roof construction. Finally the whole home object is returned. Here using the same process you can build houses with different properties.

**Note: GOF** says,

“Separate the construction of a complex object from its representation so that the same construction process can create different representations” i.e. Builder Design Pattern

**Application Areas:** Use the Builder pattern when

- The creation algorithm of a complex object is independent from the parts that actually compose the object.
- The system needs to allow different representations for the objects that are being built.

### Note

Each builder is independent of others. This improves modularity and makes the building of other builders easy. Because, each builder builds the final product step by step, we have more control on the final product.

### Sample Code:

```
// BuilderTest.java
```

```
class Cpu
```

```
{
```

```
    public String name()
```

```
{  
    return "CPU";  
}  
  
public int price()  
{  
    return 7000;  
}  
}  
  
class Monitor  
{  
    public String name()  
    {  
        return "Monitor";  
    }  
  
    public int price()  
    {  
        return 5000;  
    }  
}  
  
class Perphirals  
{  
    public String name()  
    {  
        return "Perphirals";  
    }  
}
```

```
}

public int price()
{
    return 4500;
}

}

public class BuilderTest
{
    public static void main(String[] args) throws Exception
    {
        BuilderTest bt=new BuilderTest();
        String s=bt.parts();
        int cost=bt.calPrice();
        System.out.println();
        System.out.println("Your Product Consists Of "+s);
        System.out.println();
        System.out.println("Your Product Total Cost is Rs."+cost+"/-");
    }

    public String parts()
    {
        return "One "+new Cpu().name()+" , One "+new Monitor().name()+" & Some "+
            new Perphirals().name();
    }

    public int calPrice()
```

## JAVA Means DURGA SOFT

```
{  
    return new Cpu().price()+new Monitor().price()+new Perphirals().price();  
}  
}
```

### Output

```
C:\Design patterns\Programs\Builder>javac BuilderTest.java  
C:\Design patterns\Programs\Builder>java BuilderTest  
Your Product Consists Of One CPU, One Monitor & Some Perphirals  
Your Product Total Cost is Rs.16500/-
```

## 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](http://www.durgasoftonlinetraining.com) [durgasoftonlinetraining@gmail.com](mailto:durgasoftonlinetraining@gmail.com) Ph: +91- 8885252627 +91- 7207212428

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...

**JAVA MEANS DURGASOFT**

INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED  
**DURGA**  
SOFTWARE SOLUTIONS

#202 2<sup>nd</sup> FLOOR  
[www.durgasoft.com](http://www.durgasoft.com)

040-64512786  
+91 9246212143  
+91 8096969696

JAVA Means DURGA SOFT



*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](http://www.durgasoft.com)**