

Software Design Pattern (with focus on Creational Patterns, Structural Patterns and Behavioral Patterns)

(Two day Workshop)

About the Course

In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code. This two-day course is essential for those who want to understand how this description or template can be used to solve a problem that can be used in many different situations. The course commences with a discussion on the need of design pattern and different types of patterns.

The concept of design pattern, its uses and different Creational Patterns, Structural Patterns and Behavioral Patterns are briefly discussed, together with focusing how reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.

Each design pattern is discussed with a detailed example along with relevant java code. Participants will learn how design patterns provide general solutions, documented in a format that doesn't require specifics tied to a particular problem.

The theory presented during the course is supported by a number of practical exercises and case study that demonstrate the concept of design pattern.

Workshop objectives

- ❖ A clear understanding of the role of pattern design and how this relates to the goal of overall problem solution
- ❖ Learning different Creational Patterns, Structural Patterns and Behavioral Patterns
- ❖ Understanding how reusing design patterns help to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.

Workshop Contents

The two day workshop would cover the topics as noted below. There would be exercises at appropriate junction points to elaborate and understand the concepts.

Introduction

Design Pattern Concepts

Common Design Patterns **

- **Creational Patterns**
 - Abstract Factory
 - Builder
 - Factory Method
 - Object Pool
 - Prototype
 - Singleton
- **Structural patterns**
 - Adapter
 - Bridge
 - Decorator
 - Proxy
- **Behavioral patterns**
 - Command
 - Iterator
 - Mediator
 - Observer
 - Template Method
 - Visitor

Review and Conclusion

** Each pattern will be discussed in the following format:

- Intent
- Problem
- Discussion
- Structure
- Example
- Check list
- Rules of thumb
- Code example discussion in Java
- Code generation using Star UML

Note:

- Code examples in C#, C++, Delphi, PHP would also be provided for reference purposes
- For hands-on, the course requires Eclipse for Java and any respective Interpreter and/or IDE for .Net or PHP