.NET Design Patterns with C#

Course Number 103 – 32 Hours

Overview

Design patterns are reusable solutions to common software problems encountered on while programming. This course explores various patterns, mostly from the classic "Gang of Four" (GoF) book and their implementation, while taking advantage of various features of the C# language and the .NET platform.
A brief overview of the Unified Modeling Language (UML) is discussed, to become the basis of patterns description.
The course provides labs exercises so the students can leverage the theoretical material in a more practical way.

On Completion, Delegates will be able to

- Understand what patterns are and when to use them
- Understand the basic building blocks of UML
- Implement patterns with the .NET framework and the C# language
- Use advanced C# features effectively
- Understand the various patterns and their typical use
- Understand .NET specific patterns

Who Should Attend

The course is intended for experienced C# developers who want to be exposed to design patterns and their possible implementation with .NET and C#

Prerequisites

Delegates must be experienced C# programmers (at least C# 2.0, preferably C# 3.0). Strong object oriented knowledge is assumed.
Familiarity with UML notation is a plus.

Course Contents

Module 1: Introduction to Patterns

- Object Oriented Design Overview
- OO Design Challenges
- What are design patterns?
- What is UML?
- Basic UML notation
- UML Class and Sequence diagrams
- Other UML diagrams (overview)

Module 2: Creational Patterns

- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton

Module 3: Structural Patterns

- Adapter
- Bridge
- Composite
- Decorator
- Façade
- Flyweight
- Proxy

Module 4: Behavioral Patterns

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

Module 5: .NET Specific Patterns

- Deterministic Finalization
- Declarative Programming with Custom Attributes
- Iterators
- Concurrency Patterns