

Design Patterns for C#

This Design Patterns Training will familiarize you with all the fundamental and advanced techniques of Object-Oriented Design. You will learn how to consider any Object Design situation, identify all its concepts and relationships, and model them. Furthermore, you'll learn advanced Design strategies based on the most powerful Design Patterns. It will lay the ground to implement these patterns in an Object-oriented language like C#. This is a background all professional, serious developers should have. Without it, OO development requires hard labor but brings very little reward. Sessions offer a very balanced blend of lectures and exercises; it renders the learning process attractive, interactive, and very thorough.

Objectives:

This course will familiarize you with:

- The concepts of Objects, Classes, their relationships and how to model them.
- The fundamental techniques of Object Design.
- The main Design Patterns:
 - Creational Patterns
 - Structural Patterns
 - Behavioral Patterns
- When to use Design Patterns, as opposed to "reinventing the wheel".
- The criteria to decide which pattern(s) to use.

Target Audience

- Technical Leads
- Software Architects
- Developers.

Prerequisites :

1. Course participants must be involved in development activities.
2. Must have strong understanding of OOP
3. Must have worked on C# at least for 2 years.

Course Duration: 2 Days

Day 1

Introduction, MVC, Roles, Interface-vs-Implementation

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Factory Method Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Singleton Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Abstract Factory Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"

- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Proxy Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

Day 2

The Composite Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Decorator Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Observer Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.

- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Adapter Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions

The Template Method Design Pattern

- Problem - a description of a common software engineering problem.
- Solution - a description of the "best practice" way to solve that problem.
- Setup - C# code sample that creates objects.
- Use - C# code sample that uses objects.
- Design Pattern - a description of what makes this a "design pattern"
- When To Use (this pattern)
- Makes It Easier To ...
- Reuses (previous design patterns)
- References to the literature
- Lab instructions