

Writing clean code with C#

Course Number 42731 – 16 Hours

Overview

Over the lifetime of a product, maintaining the product is actually one (if not the most) expensive area of the overall product costs. Writing clean code can significantly lower these costs, makes you more efficient during the initial development time and results in more stable code. Writing elegant and efficient code with minimal dependencies and complete error handling will make it difficult for bugs to hide and easy to maintain the code.

In this course participants will learn how to apply advanced C# techniques in order to improve the efficiency, readability, testability and extensibility of code. This course is based on C# and Visual Studio 2010.

On Completion, Delegates will be able to

- Understand the importance of writing Clean Code
- Write clean code in new designs
- Clean up legacy code
- Understand how design patterns assist in writing clean code

Prerequisites

- Programming experience with C#, using object-oriented programming techniques
- Participants should be familiar with the GOF design patterns

Course Contents

Module 1: Introduction

- Motivation
- What is clean code?
- Four Rules for Simple Design

Module 2: Choosing Names

- Use intention revealing names and avoid disinformation
- Use pronounceable and searchable names
- Avoid encodings
- Class and method names
- Avoid shortened names
- Pick one word per concept and one concept per word
- Use solution domain names and problem domain names
- Add meaningful context

Module 3: Comments and Formatting

- Comments:
 - Comments Lie
 - Bad Comments
 - Good Comments
- Formatting:
 - Is Formatting Important?
 - Vertical Formatting
 - Horizontal Formatting

Module 4: Objects and Data Structures

- Objects and Data Structures Overview
- Data Transfer Objects (DTOs)
- Hybrid structures
- The Law of Demeter

Module 5: Clean Classes, Functions and Error Handling

- Clean Classes
- Class Organization
- Cohesion
- Clean Functions
- Clean Error Handling

Module 6: Clean Systems, Tests, Build Environment

- Building Clean Systems
- Separate Construction From Use
- Dependency Injection
- Separate of Concerns
- Defining Boundaries
- Unity
- The Three Laws of TDD
- Clean Tests
- Tests should be F.I.R.S.T
- Clean Build Environment

Module 7: Measuring Code Quality

- Component Code Metrics
- Main Sequence
- The Level Metric
- Lack of Cohesion Of Methods
- Introducing NDepend