Clean Code and Code Refactoring Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Two essential skills contribute to becoming an effective software developer: **writing clean, understandable code** and **improving the internal structure of existing code without altering its externally observable behavior**.

Clean Code refers to a collection of principles and practices that make source code easier for people to read, understand, and modify. The goal is not simply to develop software that works, but to ensure that its implementation remains clear and accessible to other developers.

Writing clean code involves organizing logic consistently, establishing clear relationships between software components, and choosing meaningful names for variables, methods, and classes. When these elements are properly structured, developers can understand the purpose of the code more easily and make changes with greater confidence.

Refactoring is the process of reorganizing and improving the internal structure of software while preserving its existing functionality and external behavior. Its purpose is to reduce unnecessary complexity and make the code simpler, more understandable, maintainable, and adaptable to future changes.

Clean Code and Code Refactoring Training introduces participants to the fundamental principles and practical techniques behind both approaches. The course covers readable code development, effective naming conventions, function and method design, code reviews, identifying code smells, and applying different refactoring techniques.

Throughout the training, participants will develop the skills needed to improve existing code, reduce complexity, and make software maintenance and future development more manageable.

What You Will Learn

By the end of this training, participants will be able to:

- Apply Clean Code principles to write readable and maintainable code.
- Follow code formatting and meaningful naming conventions.
- Improve the structure and clarity of functions and methods.
- Enhance code review and documentation practices.
- Identify Code Smells and common code quality issues.
- Apply refactoring techniques without changing existing code behavior.
- Restructure methods, classes, and object relationships.
- Understand performance benchmarking and design pattern-based refactoring techniques.

Clean Code and Code Refactoring Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Outline

1. Introduction to Clean Code

- Understanding Clean Code
- Why Clean Code matters in software development
- Fundamental Clean Code principles

2. Code Formatting and Naming Conventions

- Code formatting and indentation practices
- Naming standards for variables, methods, and classes
- Improving code readability through meaningful naming

3. Functions and Methods

- Working with function arguments and return values
- Naming, organizing, and improving functions

4. Code Reviews and Documentation

- Understanding when code reviews should be performed
- Writing clear and unambiguous code comments
- Reviewing and validating code to support long-term maintainability

5. Introduction to Refactoring

- Understanding the concept of refactoring
- Why refactoring is important
- Key benefits of refactoring practices

6. Refactoring Techniques and Practices

- Detecting and eliminating code smells
- Techniques for improving overall code quality
- Practical refactoring exercises
- Performance improvement
- Performance measurement and benchmarking

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Method-level refactoring
- Refactoring classes and objects
- Reorganizing class hierarchies
- Applying pattern-based refactoring techniques

7. Code Smells

- Understanding and identifying code smells
- Common code smells in C# development
- Improving code quality through refactoring