Faysal Aziz

C# Engineering Masterclass

From expressive code to production deployment in 2 days

FA-0125 · 2 days

Software Development · DevOps, Cloud & Infrastructure

Introduction

As modern .NET evolves, C# engineers are expected to master advanced language constructs, concurrency control, rich client interfaces, robust security, and automated delivery. This two-day advanced course, led by an instructor with **over 30 years of real industry experience**, focuses on **practical, production-level development** using modern C# and .NET frameworks. Participants will learn how to build efficient, secure, and maintainable C# applications ready for real-world deployment.

Learning outcomes

Participants will be able to:

- Leverage lambda expressions and expression trees for functional programming patterns.
- Implement asynchronous operations with robust error handling and scalability.
- Design and build modern WPF desktop applications using MVVM architecture.
- Integrate security principles across application and infrastructure layers.
- Automate build, test, and deployment workflows using CI/CD pipelines.
- Connect design, security, and delivery into cohesive engineering processes.

Prerequisites

- Proficiency in C# and the .NET runtime.
- Experience with object-oriented design and asynchronous programming.

- Familiarity with WPF or other UI frameworks.
- Basic understanding of security and CI/CD concepts.

Training outline

1. Lambda Expressions and Functional Constructs

- Overview of lambda syntax and usage patterns
- Delegates, Func<>, Action<>, and expression trees
- Functional paradigms in modern C#
- Context capture and closure behavior
- Performance implications of functional constructs
- Integration of lambdas within event-driven and reactive patterns

2. Asynchronous Processing

Async programming model in .NET

- Tasks, async/await, and synchronization contexts

Concurrency management

- Threading, parallelism, and synchronization primitives

Designing asynchronous flows

- Background operations and task composition

Performance and responsiveness

- Avoiding deadlocks and managing cancellation tokens

- Testing and debugging asynchronous systems

3. Windows Presentation Foundation (WPF)

- WPF architecture and MVVM pattern
- Data binding, commands, and dependency properties
- Component-based UI design
- Asynchronous UI operations and responsiveness
- Styling, templating, and reusability

- Deployment and packaging strategies for desktop applications

4. Security Measures

- Secure coding and data handling in .NET

  Authentication and authorization frameworks

  - Role-based, claims-based, and token-based security

  Encryption and secure storage

  - Managing secrets, configurations, and sensitive data

  Secure communication

  - HTTPS, TLS, and certificate management
- Threat detection, monitoring, and audit logging

5. CI/CD Deployment Experience

Continuous integration workflows

- Automated builds, testing, and packaging

Continuous delivery and deployment

- Artifact management, environment configuration, and versioning

Build and release automation

- GitHub Actions, Azure DevOps, and Jenkins pipelines

Deployment strategies

- Blue-green and staged releases

Observability and feedback loops

- Monitoring, rollback, and incident response

6. Integration and Synthesis

- Alignment of language, architecture, and delivery pipelines
- Balancing security, performance, and maintainability
- Real-world implementation patterns and case insights
- Strategic guidance for scaling C# projects in production

© 2026 Faysal Aziz · Kuala Lumpur