SpiralTrain
developer courses

Advanced C# Programming

Audience Advanced C# Programming Course

The Advanced C# Programming course is intended for experienced C# developers who want to deepen their expertise and master modern C# language features.

Prerequisites Advanced C# Programming Course

Solid knowledge of C# and .NET, experience with object-oriented programming, and familiarity with basic asynchronous programming concepts are required.

Training Execution Advanced C# Programming

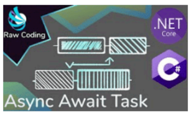
The training consists of interactive theory sessions combined with live demonstrations and extensive hands-on labs under the guidance of an experienced trainer.

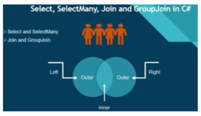
Certificate Advanced C# Programming

Participants will receive a certificate of completion for the Advanced C# Programming course after successfully completing the training.

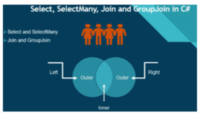
Duration: 4 days **Price: € 2999**

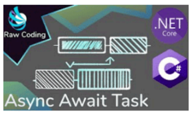
Open Schedule





Advanced C# Programming





Content Course Advanced C# Programming

In the Advanced C# Programming course, participants deepen their knowledge of modern C# features such as asynchronous programming, LINQ, dependency injection, reflection, and performance optimization.

Advanced C# Language Features

The course starts by exploring advanced language features such as nullable reference types, records, enhanced pattern matching, local functions, dynamic types, and advanced string handling.

Delegates, Events and Lambdas

Participants dive into delegates, event handling, lambda expressions, multicast delegates, expression trees, asynchronous delegates, and practical usage patterns.

Generics and Collections

The focus is on using generics with constraints, covariance and contravariance, creating custom collections, working with immutable and concurrent collections, and advanced dictionary techniques.

LINQ and Functional Programming

Participants learn LINQ query and method syntax, deferred execution, custom operators, and the basics of functional programming including immutability and higher-order functions.

Asynchronous and Parallel Programming

This module covers async/await, Task Parallel Library (TPL), ValueTask optimizations, cancellation tokens, deadlock prevention, and working with IAsyncEnumerable for async streams.

Dependency Injection and Testing

Participants learn about dependency injection patterns, mocking and testing with frameworks like Moq, applying TDD, and managing services correctly in .NET Core applications.

Reflection, Attributes and Dynamic Programming

This module focuses on reflection, dynamic loading of assemblies, working with attributes, dynamic object creation, late binding, and practical use cases for reflection.

Memory Management and Performance

Participants gain insight into garbage collection internals, the dispose pattern, memory profiling, object pooling, stackalloc optimization, and improving async performance.

Advanced Topics and Best Practices

Finally, the course covers exception handling best practices, logging, using source generators, global using statements, native code interoperability, and writing clean, maintainable code.

Modules Course Advanced C# Programming

Module 1: Advanced C# Features	Module 2: Delegates and Lambdas	Module 3: Generics
Nullable Reference Types	Delegates Deep Dive	Generic Methods and Classes
Pattern Matching Enhancements	Multicast Delegates	Generic Constraints
Records and Value-Based Equality	Events and Event Handling	Covariance and Contravariance
Tuples and Deconstruction	Anonymous Methods	Collections Overview
Local Functions	Lambda Expressions	IEnumerable, ICollection, IList
Target-Typed New Expressions	Expression Trees	Custom Generic Collections
Default Interface Methods	Func, Action, Predicate	Immutable Collections
Dynamic Types	Event Aggregator Pattern	Concurrent Collections
Using Span and Memory	Asynchronous Delegates	Collection Initializers
Advanced String Manipulation	EventHandler Usage	Advanced Dictionary Usage
Module 4: Functional Programming	**Module 5: Asynchronous Programming**	**Module 6: Dependency Injection**
Introduction to LINQ	Async and Await Deep Dive	Principles of DI
Query Syntax vs Method Syntax	Task Parallel Library (TPL)	Service Lifetimes
LINQ to Objects	ValueTask and Memory Optimization	Constructor Injection
Deferred vs Immediate Execution	Parallel.For and Parallel LINQ	Method Injection
Custom LINQ Operators	Cancellation Tokens	Property Injection
Func and Action in LINQ	Exception Handling in Async Code	Mocking and Stubbing
Functional Programming Concepts	SynchronizationContext	Using Moq and NSubstitute
Immutability in C#	Deadlocks and Best Practices	Unit Testing Best Practices
Higher-Order Functions	Channels and Dataflow	Test-Driven Development
Pipeline and Fluent APIs	IAsyncEnumerable and Async Streams	DI in .NET Core
Module 7: Dynamic Programming	**Module 8: Memory Management**	**Module 9: Advanced Topics**
Reflection Basics	Garbage Collection in Depth	Best Practices for Exceptions
Loading Assemblies Dynamically	Dispose Pattern and IDisposable	Logging and Monitoring
Reading Metadata	Memory Profiling Tools	Code Analysis Tools (Roslyn)
Custom Attributes	Avoiding Memory Leaks	Source Generators
Attribute Usage and Creation	Object Pooling	Global Using Statements
Dynamic Object Manipulation	Stackalloc and Span for Performance	Nullable Contexts Best Practices
ExpandoObject	Performance Profiling	Minimal APIs
Dynamic Keyword	Async Performance Optimization	Working with Files and Streams
Late Binding	BenchmarkDotNet for Microbenchmarks	Native Code Interoperability
Reflection Scenarios	High-Performance C# Coding	Writing Clean Code