



☎ +44 7989 401397

✉ info@olsensoft.com

Advanced C# Development

(4 days)

Course overview

Once you've mastered the core features of C# as a programming language, you're ready to take the next step. The .NET Framework offers an incredibly rich and diverse set of APIs that cover all aspects of contemporary development. This course takes a detailed look at the areas of C# and .NET that have particular resonance to developers today, including asynchrony, creating decoupled and adaptable systems, test-driven development, .NET Core features, and containerization using Docker. The course also covers all the new features in C# 8.

What you'll learn

- Asynchronous programming techniques
- Parallelization and concurrency
- Debugging multithreaded code
- Reflection and metadata
- .NET Core and ASP.NET Core features
- Containerization using Docker
- Dynamic programming
- Dependency injection
- Test-driven development

Prerequisites

- At least 6 months C# programming experience

Course details

- **Delegates, Events, and Lambda Expressions:** Delegates; Anonymous Methods and Lambdas; Standard Delegate Types; Events; Multicast Delegates; Asynchronous Delegates
- **Generics in Depth:** Generics Essentials; A Closer Look at Generics; Generic Structs, Interfaces, Delegates; Generic Methods
- **Additional C# Language Features:** Tuples; Initialization Techniques; Anonymous Types; Extension Methods
- **LINQ to Objects:** Introduction to LINQ; Using LINQ with Arrays; Using LINQ with Collections; LINQ Techniques; LINQ Under the Hood
- **.NET Core Applications:** Getting Started with .NET Core; Overview of Containerization and Docker; Understanding Docker Images; A Closer Look at Images and Containers; How to Containerize a .NET Core Application

- [ASP.NET Core Web Applications](#): Getting Started with ASP.NET Core; Creating a Minimal ASP.NET Core Web App; Containerizing an ASP.NET Core Web App; Using VS Support for Containerization
- [What's New in C# 8](#): Nullable Reference Types; Improved Pattern Matching; Miscellaneous Language Features; Asynchronous Streams
- [Parallelization and Asynchronous Methods](#): Parallelization; Async Methods
- [Dynamic Programming](#): Using the dynamic Keyword; Dynamic Language Runtime (DLR)
- [Introduction to Design Patterns](#): Using Design Patterns; Pattern Classification; Creational Patterns; Structural Patterns; Behavioural Patterns
- [Introduction to Unit Testing in .NET](#): Setting the Scene; TDD using Test Explorer; Additional Techniques
- [Unit Testing with xUnit.net](#): Getting Started with xUnit.net; A Closer Look at xUnit.net
- [Test-Driven Development](#): Effective Test-Driven Development; Refactoring Techniques; Types of Test
- [Test Doubles](#): Making Code Testable; Overview of Test Doubles; Using Stubs; Using Mocks
- [Dependency Injection](#): DI concepts; Tools for implementing DI; Overview of Unity; Resolving dependencies; Designing for DI