

Advanced C# Programmeren

Doelgroep Cursus Advanced C# Programming

De cursus Advanced C# Programming is bedoeld voor ervaren C# developers die hun kennis willen verdiepen en optimaal gebruik willen maken van de nieuwste features in moderne C# versies.

Voorkennis Cursus Advanced C# Programming

Grondige kennis van C# en .NET, ervaring met objectgeoriënteerd programmeren en basiskennis van asynchrone programmeertechnieken zijn vereist.

Uitvoering Training Advanced C# Programming

De training combineert interactieve theorie met live demonstraties en uitgebreide hands-on labs onder begeleiding van een ervaren trainer.

Certificaat Advanced C# Programming

Na het succesvol afronden van de cursus ontvangen deelnemers een certificaat van deelname aan de cursus Advanced C# Programming.

Duur: 4 dagen

Prijs: € 2999

Open Rooster



Advanced C# Programming



Inhoud Cursus Advanced C# Programmeren

In de cursus Advanced C# Programming verdiepen deelnemers hun kennis van moderne C# features zoals asynchrone programmeren, LINQ, dependency injection, reflection en prestatieoptimalisatie.

Advanced C# Language Features

De cursus begint met geavanceerde taalfeatures zoals nullable reference types, records, pattern matching, lokale functies, dynamic types en geavanceerde stringmanipulaties.

Delegates, Events en Lambdas

In deze module verdiepen deelnemers zich in delegates, events, lambda-expressies, multicast delegates, expression trees en het gebruik van asynchrone delegates.

Generics en Collecties

De focus ligt op het gebruik van generics, constraints, covariantie en contravariantie, aangepaste collecties, immutable en concurrent collections en geavanceerd gebruik van dictionaries.

LINQ en Functioneel Programmeren

Deelnemers leren over LINQ query- en methodsyntax, deferred execution, custom operators, en de basisprincipes van functioneel programmeren zoals immutability en hogere-orde functies.

Asynchrone en Parallele Programmeren

Deze module behandelt async/await, de Task Parallel Library (TPL), ValueTask optimalisaties, cancellation tokens, deadlock voorkomen en het gebruik van IEnumerableAsync.

Dependency Injection en Testen

Deelnemers leren dependency injection patronen, mocken en testen met frameworks als Moq, het toepassen van TDD en het correct beheren van services in .NET Core applicaties.

Reflectie, Attributen en Dynamisch Programmeren

In deze module wordt gewerkt met reflection, het dynamisch laden van assemblies, het gebruik van attributen, het creëren van dynamische objecten en late binding.

Geheugenbeheer en Prestatieoptimalisatie

Deelnemers krijgen inzicht in garbage collection, dispose patterns, geheugenprofieling, object pooling, stackalloc gebruik en het optimaliseren van asynchrone code.

Geavanceerde Onderwerpen en Best Practices

Tot slot behandelt deze module exception handling best practices, logging, gebruik van source generators, globale usings, interoperabiliteit met native code en het schrijven van onderhoudbare code.

Modules Cursus Advanced C# Programmeren

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