

Advanced .NET with C#

Course 303 – 40 Hours

Overview

This course provides in-depth coverage of advanced C# and CLR mechanisms, allowing experienced C# developers harness the full power of the C# language and the CLR.

Course Objectives

Upon completion students should be able to:

- Use reflection and metadata
- Understand and use generic types effectively
- Manage memory and non-memory resources effectively
- Understand and use delegates and events
- Understand serialization mechanisms
- Understand what are AppDomains and how to use them
- Understand the connection between AppDomains, processes and threads
- Build effective multithreading applications
- Use thread synchronization primitives effectively
- Use C# 3.0 4.0 and 5.0 new features and enhancements
- Use Introp techniques between C# and native c++code
- Understand the usage of the C++ CLI language

Who Should Attend

The course is intended for experienced C# developers who want to upgrade their skills and deepen their understanding of the C# language and the .NET platform.

Prerequisites

Basic C# knowledge is a must. At least 6 months of .NET development is required.

Course Contents

Reflection and Code Generation

- Reflection basics and the Type class
- Creating instances dynamically
- Getting and setting data using reflection
- Creating and using Custom Attributes
- Code generation basics
- Using the CodeDOM
- Using IL Generator

Generics

- The need for generics
- Writing and using generic types
- Generic methods, interfaces and delegates
- Applying constraints
- Generic collections
- Other aspects of generic programming
- Nullable types

Managing Resources

- Garbage collection and its impacts
- Finalizers
- The 'Dispose' pattern
- The using statement
- The GCHandle type
- Weak references
- Resurrection, generations, large object heap
- Monitoring garbage collection
- GC types

Advanced Delegates and Events

- Working with delegates
- Creating your own delegate types
- Events
- Anonymous delegates
- Asynchronous invocation patterns
- Late binding using delegates

Serialization

- Serialization Scenarios
- Serialization Attributes
- Object Graph
- Serialization Process
- Deserialization Example
- Custom Serialization
- Other serializers

Processes, AppDomains and Threads

- Processes
- Application Domains vs. processes
- Threads and AppDomains
- AppDomains and objects
- Crossing AppDomains
- Marshalling and serialization

Multithreading

- Managed vs. Unmanaged threads
- Thread scheduling
- Creating and managing threads
- The Thread class
- Thread synchronization
- Synchronization kernel objects
- Multithreading best practices

TPL

- Parallel vs Asynchronous scenarios
- From Thread to Task
- Data parallelism
- Pliq
- Anych Await Pattern

Advanced Language Constructs

- Partial types and partial methods
- Iterators
- Expression Trees
- C# 3.0 basic features
- Extension methods
- LINQ

Interoperability

- Interoperability scenarios
- Platform Invoke (P/Invoke)
- Marshalling parameters and types
- Using COM Components
- COM Apartments and AppDomains
- Best practices

Unsafe code and C++/CLI

- What is unsafe code?
- The unsafe C# keyword
- Unsafe code rules
- What is C++/CLI?
- C++/CLI basics
- Interoperability with C++/CLI