

Extreme .NET 4/4.5 with C#

Course Number 4151 – 40 Hours

Overview

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

On Completion, Delegates will be able to

- Understand reflection and its uses
- Use the CodeDOM to generate dynamic assemblies
- 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 the new Task Parallel Library effectively
- Use thread synchronization primitives effectively
- Create and Use iterators effectively
- Use C# 3.0 & C# 4.0 features effectively
- Understand the LINQ family of technologies and how to use it

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 Local Storage (TLS)
- Thread synchronization
- Synchronization kernel objects
- The Thread Pool
- Multithreading best practices

Tasks

- Introduction to Tasks
- Creating tasks
- Cancelling tasks
- Tasks & Exceptions
- Task schedulers
- The Parallel class
- Parallel LINQ
- Concurrent Collections

Advanced Language Constructs

- Partial types and partial methods
- Iterators
- Expression Trees
- C# 3.0 basic features
- Extension methods
- LINQ
- LINQ Operators
- Introduction to LINQ to SQL
- LINQ to XML

C# 4.0 & C# 5.0

- Introduction
- Trends
- Optional and Named Arguments
- Dynamic typing
- Improved COM interop
- Generic Variance
- Asynchronous Programming with C# 5.0



Appendix A: Interoperability

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