.NET Framework Using C#

Course ID #: 7000-663-ZZ-Z

Hours: 28

Course Content

Course Description:

This four-day course is designed to provide a sound introduction to the .NET Framework for programmers who already know the C# language and the fundamentals of Windows Forms. It is current to Visual Studio 2019, which now includes support for cross-platform development using .NET Core. The course focuses on core portions of the .NET Framework that are common across many application areas. Separate courses are available in specific areas, such as ADO.NET, XML Programming, Windows Presentation Framework, Windows Communications Framework and ASP.NET.

Learning Objectives

- Gain a thorough understanding of the philosophy and architecture of .NET
- Acquire a working knowledge of the .NET programming model and .NET Security
- Implement multi-threading effectively in .NET applications
- Learn how to implement database applications using ADO.NET and LINQ
- Learn how to debug .NET applications using .NET diagnostic classes and tools

Prerequisites:

The student should be an experienced application developer or architect with a working knowledge of C#, including building simple GUIs with Windows Forms.

Topics:

.NET Fundamentals

- What is Microsoft .NET?
- Common Language Runtime
- CLR Serialization
- Attribute-Based Programming
- Interface-Based Programming
- Metadata
- Common Type System
- Framework Class Library
- Language Interoperability
- Managed Code
- Assemblies and Deployment
- Web Services
- ASP.NET
- Performance
- .NET Native
- .NET Core and Cross-platform Development
- XML Serialization

Class Libraries

- Components in .NET
- Building Class Libraries at the Command Line
- Class Libraries Using Visual Studio
- Using References

.NET Framework Using C#

Course ID #: 7000-663-ZZ-Z

Hours: 28

Assemblies, Deployment and Configuration

- Assemblies
- Private Assembly Deployment
- Shared Assembly Deployment
- Configuration Overview
- Configuration Files
- Programmatic Access to Configuration
- Using SDK Tools for Signing and Deployment
- Application Settings

Metadata and Reflection

- Metadata
- Reflection
- Late Binding

I/O and Serialization

- Directories
- Files
- Serialization
- Attributes

.NET Programming Model

- Memory Management and Garbage Collection
- Asynchronous Delegates
- BackgroundWorker
- Application Domains

.NET Threading

- Threading Fundamentals
- ThreadPool
- Foreground and Background Threads
- Synchronization
- Task Parallel Library

.NET Security

- Authentication and Authorization
- Code Access Security
- Sandboxing
- Permissions
- Role-Based Security
- Principals and Identities

Interoperating with COM and Win32

- .NET Client Calling a COM Server
- 64-bit System Considerations
- PInvoke

.NET and LINQ

- ADO.NET Overview
- .NET Data Providers
- Connections
- Using LocalDB
- Commands
- DataReaders and Connected Access
- Data Sets and Disconnected Access
- Language Integrated Query

11. Debugging Fundamentals

- Compile-time Errors and Run-time Errors
- Configuring Debug, Release, and Special Builds
- Visual Studio Debugger
- Just-In-Time Debugging
- Attaching Debugger to a Running Process

12. Tracing

- Tracing
- Event Logs

13. More About Tracing

- Using the BooleanSwitch and TraceSwitch Classes
- Print Debugging Information with the Debug Class
- Instrumenting Release Builds with the Trace Class
- Using Listeners
- Implementing Custom Listeners

Appendix A: .NET Remoting

- Marshal by Value
- Marshal by Reference
- .NET Remoting