



BUILDING TRAINING SOLUTIONS
FOR THE IT WORLD

Advanced .NET Framework Programming

Days: 5
Format: Instructor-Led
Class Code: CH-5657
Certification Exams: 70-536
Certification Track: MCTS

Recommended Course Sequence

Knowledge of prerequisites
noted below.

*Course content is subject to change
without notice.*

Course Description:

This five-day instructor-led course provides students with the knowledge and skills to program Microsoft .NET Framework applications by using Microsoft Visual Studio 2005 or Visual Studio 2008 development system. This course helps students to prepare for Exam 70-536 Microsoft .NET Framework - Application Development Foundation, which is a core requirement for any of the Visual Studio MCTS certifications.

Target Student:

The audience for this course consists of application developers with the skills to develop business applications by using Visual Studio 2005/2008 and either Microsoft Visual Basic development system or Microsoft Visual C# development tool.

Prerequisites:

Before attending this course, students must meet the following prerequisites:

- Understand the purpose and components of the .NET Framework and the common language runtime.
- Understand and use the .NET Framework common type system (CTS).
- Understand basic language syntax for decision structures, loop structures, and variables.
- Write code by using language-specific functionality such as the My. classes for Visual Basic.
- Understand and use classes, objects, methods, properties, and functions.
- Write code to implement overridden methods, static (Visual C#) or Shared (Visual Basic) methods, and properties.
- Use type conversions and text conversions.
- Create and use solutions and projects by using Visual Studio 2005 or Visual Studio 2008.
- Use the Visual Studio object browser and the Visual Studio help system.

Delivery Method:

Instructor led, group-paced, classroom-delivery learning model with structured hands-on activities.

Advanced .NET Framework Programming

At Course Completion:

Upon successful completion of this course, students will be able to:

- Develop applications that use types and standard contracts.
- Manage common data by using collections.
- Deploy and configure assemblies.
- Monitor and debug applications.
- Read and write files.
- Serialize data.
- Enhance user interfaces by using System.Drawing.
- Work with cultures by using System.Globalization.
- Process text by using regular expressions and encodings.
- Encrypt and hash data by using cryptography.
- Secure code execution and resources.
- Explain how to implement application interoperability.
- Describe and use reflection, metadata, and emitting objects.
- Describe and use services, threading, and application domains.

Course Outline

Module 1: Developing Applications by Using Types and Standard Contracts

This module describes the differences between reference types and value types. The module also describes how to create generic data types, implement standard .NET Framework interfaces, use delegates and events, and use attributes and exceptions.

Lessons
<ul style="list-style-type: none">■ Explaining Value Types and Reference Types.■ Working with Generic Types.■ Implementing .NET Standard Contracts.■ Implementing Delegates and Events.■ Working with Attributes and Exceptions.
Lab: Developing Applications by Using Types and Standard Contracts
<ul style="list-style-type: none">■ Using Nullable Types.■ Defining a Generic Type.■ Implementing Standard .NET Framework Interfaces.■ Throwing and Catching Exceptions (If Time Permits).■ Raising and Handling Events (If Time Permits).

Advanced .NET Framework Programming

Module 2: Managing Common Data by Using Collections

This module introduces the object-based collection classes in the System.Collections namespace. The module also describes the generic collection classes in the System.Collections.Generic namespace and outlines the benefits of the generic collections. The module also covers several specialized collections in the System.Collections.Specialized namespace.

Lessons
<ul style="list-style-type: none">Working with Object-Based Collections.Working with Generic Collections.Working with Specialized Collections.
Lab: Managing Common Data by Using Collections
<ul style="list-style-type: none">Using the Dictionary Generic Collection.Using the List Generic Collection.Using the NameValue Collection Specialized Collection.

Module 3: Deploying and Configuring Assemblies

This module describes the key features of assembly configuration and installation. It also explains how to install assemblies and how to configure assemblies and the .NET Framework.

Lessons
<ul style="list-style-type: none">Installing and Configuring Assemblies.Installing Assemblies by Using the Installer.Configuring Assemblies.Configuring the .NET Framework.
Lab: Deploying and Configuring Assemblies
<ul style="list-style-type: none">Managing the Configuration Settings of an Assembly.Deploying an Application by Using Windows Installer.

Module 4: Monitoring and Debugging Applications

This module describes how to use the available classes of the System.Diagnostics namespace to monitor and debug a .NET Framework application.

Lessons
<ul style="list-style-type: none">Working with Application Processes.Managing Application Performance.Reading and Writing to an Event Log.Debugging and Tracing Applications.
Lab: Monitoring and Debugging Applications
<ul style="list-style-type: none">Monitoring Application Performance.Logging Information in a Custom Event Log.Adding and Configuring Tracing Statements in an Application.

Advanced .NET Framework Programming

Module 5: Reading and Writing Files

This module describes how to use the classes in the .NET Framework Class Library to manipulate files and folders and how to read and write data to and from files and memory.

Lessons
<ul style="list-style-type: none">■ Managing the File System.■ Reading and Writing Data by Using Streams.■ Compressing and Protecting Data by Using Streams.■ Improving Application Security by Using Isolated Storage.
Lab: Reading and Writing to File and Folders
<ul style="list-style-type: none">■ Archiving Files.■ Compressing Files.■ Storing and Retrieving User Preferences.

Module 6: Serializing Data

This module describes how the .NET Framework Class Library implements serialization and explains how to use the classes provided to serialize and deserialize classes and structures. This module also describes how to customize the serialization mechanism implemented by the .NET Framework.

Lessons
<ul style="list-style-type: none">■ Serializing and Deserializing Objects by Using Runtime Serialization.■ Customizing the Runtime Serialization and Deserialization Processes.■ Serializing and Deserializing Objects As XML Data.
Lab: Serializing Data
<ul style="list-style-type: none">■ Serializing and Deserializing Data Across a Network by Using Runtime Serialization.■ Customizing the Runtime Serialization Process.■ Serializing and Deserializing Data as XML.

Module 7: Enhancing User Interfaces by Using System.Drawing

This module describes the key features of the System.Drawing namespace that the .NET Framework provides. It also explains how to create and modify your own custom drawings.

Lessons
<ul style="list-style-type: none">■ Drawing Fundamentals.■ Drawing Lines and Shapes.■ Rendering Bitmaps and Icons.
Lab: Drawing to a Windows Form
<ul style="list-style-type: none">■ Drawing a Feedback Bar.■ Drawing a Feedback Pie Chart.■ Implementing an Automatic Double Buffer.■ Adding Fonts to Your Application.■ Saving Your Scaled Image.

Advanced .NET Framework Programming

Module 8: Working with Cultures by Using System.Globalization

This module describes how to use the System.Globalization namespace to work with culture information and perform culture-sensitive string comparisons. It also describes how to create a custom culture.

Lessons
<ul style="list-style-type: none"> ■ Working with Culture Information ■ Formatting and Sorting Culture-Sensitive Data ■ Creating a Custom Culture.
Lab: Working with Cultures by Using System.Globalization
<ul style="list-style-type: none"> ■ Managing Culture Information. ■ Creating a Custom Culture.

Module 9: Processing Text by Using Regular Expressions and Encodings

This module describes the key features of the System.Text namespace that the .NET Framework provides. It explains how to store and manipulate strings, how and when to implement regular expressions, and how to customize encodings to produce the correct results when you process text

Lessons
<ul style="list-style-type: none"> ■ Handling Text and Large Strings. ■ Using Regular Expressions. ■ Encoding Text.
Lab: Processing Text by Using Regular Expressions and Encodings
<ul style="list-style-type: none"> ■ Handling Text and Strings. ■ Creating and Using Regular Expressions. ■ Working with Encoding.

Module 10: Encrypting and Hashing Data by Using Cryptography

This module describes when to use data encryption and hashing and explains how to use the classes in the .NET Framework to perform these cryptographic tasks. It also discusses how to customize the implementation of specific algorithms by extending base classes in the System.Security.Cryptography namespace.

Lessons
<ul style="list-style-type: none"> ■ Working with Encryption and Hashing. ■ Encrypting and Decrypting Data. ■ Hashing Data. ■ Extending Cryptography.
Lab: Creating a Cryptographic Application
<ul style="list-style-type: none"> ■ Creating an Asymmetric Key. ■ Encrypting a File. ■ Decrypting a File. ■ Exporting and Importing a Public Key. ■ Getting a Private Key.

Advanced .NET Framework Programming

Module 11: Securing Code Execution and Resources

This module describes how code access security (CAS) works, how to make CAS permission checks in code, and how to configure CAS security policy. It also introduces Windows operating system access checks and explains how to use the .NET Framework base class library to read and modify Windows access control lists (ACLs). This module also describes how .NET Framework security performs authorization checks and how to customize the mechanism.

Lessons
<ul style="list-style-type: none">■ Using Code Access Security.■ Securing Code Execution by Using Policy.■ Securing Resources by Using Access Control.■ Customizing Authentication and Authorization.
Lab: Custom Authentication by Using Principal Objects
<ul style="list-style-type: none">■ Viewing the Starter Solution.■ Adding Role-Based Security.■ Testing the Application.

Module 12: Application Interoperability

This module describes how to invoke functions that are implemented in unmanaged DLLs, and how to use various techniques for integrating Component Object Model (COM) components into managed applications. It also explains how to make managed components that are built by using the .NET Framework available to unmanaged COM client applications

Lessons
<ul style="list-style-type: none">■ Using the Platform Invoke Service.■ Integrating COM Components into a .NET Framework Application.■ Integrating Managed Components into an Unmanaged Application.
Lab: Application Interoperability
<ul style="list-style-type: none">■ Integrating Unmanaged Functions into a Managed Application.■ Integrating a COM Component into a Managed Application by Creating an Interop Assembly.■ Integrating a COM Component into a Managed Application by Using Late Binding.■ Manually Creating an Interop Assembly for a COM Component.

Advanced .NET Framework Programming

Module 13: Reflection, Metadata, and Emitting Objects

This module describes how to use the classes in the .NET Framework class library to examine a program, alter the behavior or structure of the program as it runs, and create and run new code.

Lessons
<ul style="list-style-type: none">■ Reflecting on Objects.■ Adding Assembly Metadata.■ Emitting Objects by Using Builder Classes.
Lab: Creating an Add-in Framework by Using Reflection
<ul style="list-style-type: none">■ Creating a Custom Attribute to Decorate Add-in Classes.■ Using Reflection to Discover the Add-in Classes.■ Creating a Mapping from Columns to Properties.■ Creating a Dynamic Method to Set a Property Value.■ Invoking the Row Handler.

Module 14: Services, Threading, and Application Domains

This module describes how to use the Microsoft .NET Framework classes to write Windows services and classes to install services. It also explains how to use the .NET Framework classes to create thread objects to execute code and to synchronize execution between threads. Finally, this module describes how to access and configure application domains.

Lessons
<ul style="list-style-type: none">■ Creating and Installing Windows Services.■ Creating Multithreaded Applications.■ Manually Working with Application Domains.
Lab: Creating a Windows Service Application
<ul style="list-style-type: none">■ Creating a Service Project.■ Creating the Installation Project.■ Creating a Client Application.■ Writing the Service Code