

Programming with the Microsoft .NET Framework (Microsoft Visual Basic .NET)

Introduction

The goal of this course is to help application developers understand the Microsoft .NET Framework. In addition to offering an overview of the .NET Framework and an introduction to key concepts and terminology, the course provides a series of labs, which introduce and explain .NET Framework features that are used to code, debug, tune, and deploy applications.

Pre-Requisites

Before attending this course, students must be proficient in the Microsoft Visual Basic programming language and have been exposed to the Visual Basic .NET language. Students can meet these prerequisites by taking Course 2559: Introduction to Visual Basic .NET Programming.

Outcomes

After completing this course, students will be able to:

- List the major elements of the .NET Framework and explain how they fit into the .NET platform, and explain the main concepts behind the common language runtime and use the features of the .NET Framework to create a simple application.
- Create and use components in Microsoft Windows Forms-based and ASP.NET-based applications.
- Use the deployment and versioning features of the .NET runtime to deploy multiple versions of a component.
- Create, use, and extend types by understanding the common type system architecture.
- Create classes and interfaces that are functionally efficient and appropriate for given programming scenarios.
- Use the .NET Framework class library to efficiently create and manage strings, arrays, collections, and enumerators.
- Use delegates and events to make an event sender signal the occurrence of an action to an event receiver.
- Control how memory and other resources are managed in the .NET Framework.
- Read from and write to data streams and files.
- Use the basic request/response model to send and receive data over the Internet.
- Serialize and deserialize an object graph.
- Create distributed applications using XML Web services and object remoting.

Course Details

Course code: MS 2415

Duration: 5 days

Starting time: 9.00 am

Finishing time: 4.30 pm

Lunch and refreshments are provided.

Booking guidelines

Contact our Learning Consultants on 1300 86 87246 and we will assist you with your booking.



(1300 86 87246
1300 TO TRAIN

Course Outline

Ø Module 1: Overview of the Microsoft .NET Framework

The following topics are covered in this module:

Lessons

- Overview of the Microsoft .NET Framework
- Overview of Namespaces

After completing this module, you will be able to list the major elements of the .NET Framework, including:

- Describing the .NET Framework and its components.
- Explaining the relationship between the .NET Framework class library and namespaces.

Ø Module 2: Introduction to a Managed Execution Environment

The following topics are covered in this module:

Lessons

- Writing a .NET Framework Application
- Compiling and Running a .NET Framework Application

After completing this module, you will be able to explain the main concepts behind the common language runtime and use the features of the common language runtime to create a simple application, including:

- Creating simple console applications in Visual Basic .NET.
- Explaining how code is compiled and executed in a managed environment.
- Explaining the concept of garbage collection.

Ø Module 3: Working with Components

The following topics are covered in this module:

Lessons

- An Introduction to Key .NET Framework Development Technologies
- Creating a Simple .NET Framework Component
- Creating a Simple Console Client
- Creating an ASP.NET Client

After completing this module, you will be able to create and use components in Windows Form-based and ASP.NET-based application, including:

- Creating a simple .NET Framework component in Visual Basic.
- Implementing structured exception handling.
- Creating a simple .NET Framework console application that calls a component.
- Creating a .NET Framework client application by using the Windows Forms library.
- Creating an .ASP.NET page that uses the previously developed .NET Framework component to create an ASP.NET application.

Ø Module 4: Deployment and Versioning

The following topics are covered in this module:

Lessons

- Introduction to Application Deployment
- Application Deployment Scenarios
- Related Topics and Tools

After completing this module, you will be able to use the deployment and versioning features of the .NET common language runtime to deploy multiple versions of a component, including:

- Packaging and deploying simple and componentized applications.
- Creating strong-named assemblies.

- Installing and removing assemblies in the global assembly cache.
- Configuring applications to control binding based on assembly location and version data.

Ø Module 5: Common Type System

The following topics are covered in this module:

Lessons

- Introduction to the Common Type System
- Elements of the Common Type System
- Object-Oriented Characteristics

After completing this module, you will be able to create, use, and extend types, including:

- Describing the difference between value types and reference types.
- Explaining the purpose of each element in the type system, including values, objects, and interfaces.
- Explaining how the object-oriented programming concepts such as abstraction, encapsulation, inheritance, and polymorphism, are implemented in the common type system.

Ø Module 6: Working with Types

The following topics are covered in this module:

Lessons

- System.Object Class Functionality
- Specialized Constructors
- Type Operations
- Interfaces
- Managing External Types

After completing this module, you will be able to create classes and interfaces that are functionally efficient and appropriate for given

Course Outline

programming scenarios, including:

- Applying attributes to control visibility and inheritance in classes and interfaces.
- Creating and using interfaces that define methods and properties.
- Explaining how boxing and unboxing work and when boxing and unboxing occur.
- Using operators to determine types at run time and to cast values to different types.
- Explaining what features are available to work with unmanaged types, such as COM types.

Ø Module 7: Strings, Arrays, and Collections

The following topics are covered in this module:

Lessons

- Strings
- Collections Defined
- .NET Framework Arrays
- .NET Framework Collections

After completing this module, you will be able to use the .NET Framework class library to create and manage strings, arrays, collections, and enumerators, including:

- Parsing, formatting, manipulating, and comparing strings.
- Using the classes in the System.Array and System.Collections namespaces.
- Improving the type safety and performance of collections by using specialized collections and class-specific code.

Ø Module 8: Delegates and Events

The following topics are covered in this module:

Lessons

- Delegates
- Multicast Delegates

- Events
- When to Use Delegates, Events, and Interfaces

After completing this module, you will be able to use delegates and events to cause an event sender to signal the occurrence of an action to an event receiver, including:

- Using the delegate class to create type-safe callback functions and event-handling methods.
- Using the Event keyword to simplify and improve the implementation of a class that raises events.
- Implementing events that conform to the .NET Framework guidelines.

Ø Module 9: Memory and Resource Management

The following topics are covered in this module:

Lessons

- Memory Management Basics
- Non-Memory Resource Management
- Implicit Resource Management
- Explicit Resource Management
- Optimizing Garbage Collection

After completing this module, you will be able to describe and control how memory and other resources are managed in the .NET Framework, including:

- Describing how garbage collection manages object memory.
- Implicitly managing non-memory resources by using a destructor's finalize code.
- Explicitly managing non-memory resources by using client-controlled deterministic release of resources.
- Writing code by using the temporary resource usage design pattern.
- Programmatically controlling the behavior of the garbage collection.

- Describing advanced garbage collection features.

Ø Module 10: Data Streams and Files

The following topics are covered in this module:

Lessons

- Streams
- Readers and Writers
- Basic File IO

After completing this module, you will be able to read from and write to data streams, files, and the Internet, including:

- Using Stream objects to read and write bytes to backing stores, such as strings and files.
- Using BinaryReader and BinaryWrite objects to read and write primitive types as binary values.