

Module 1: Overview of the Microsoft .NET Framework

The following topics are covered in this module:

- 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. This includes:

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

Module 2: Introduction to a Managed Execution Environment

- Writing a .NET Application
-
- Compiling and Running a .NET 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. This includes:

- Creating simple console applications in C#.
- Explaining how code is compiled and executed in a managed execution environment.
- Explaining the concept of garbage collection.

Module 3: Working with Components

The following topics are covered in this module:

- 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 applications. This includes:

- Creating a simple .NET Framework component in C#.
- 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.

Module 4: Deployment and Versioning

The following topics are covered in this module:

- 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. This includes:

- Packaging and deploying simple and componentized applications.
- Creating strong-named assemblies.
- Installing and removing assemblies from the global assembly cache.

- Configuring applications to control binding based on assembly location and version data.

Module 5: Common Type System

The following topics are covered in this module:

- An 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. This includes:

- 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 object-oriented programming concepts, such as abstraction, encapsulation, inheritance, and polymorphism, are implemented in the Common Type System.

Module 6: Working with Types

The following topics are covered in this module:

- 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 programming scenarios. This includes:

- 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.

Module 7: Strings, Arrays, and Collections

The following topics are covered in this module:

- Strings
- Terminology - Collections
- .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. This includes:

- 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.

Module 8: Delegates and Events

The following topics are covered in this module:

- 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 have an event sender object signal the occurrence of an action to an event receiver object. This includes:

- 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.

Module 9: Memory and resource Management

The following topics are covered in this module:

- 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. This includes:

- 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.

Module 10: Data Streams and Files

The following topics are covered in this module:

- Streams
- Readers and Writers
- Basic File I/O

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

- Using Stream objects to read and write bytes to backing stores, such as strings and files.
- Using BinaryReader and BinaryWriter objects to read and write primitive types as binary values.
- Using StreamReader and StreamWriter objects to read and write characters to a stream.
- Using StringReader and StringWriter objects to read and write characters to strings.
- Using Directory and DirectoryInfo objects to create, move, and enumerate through directories and subdirectories.
- Using FileSystemWatcher objects to monitor and react to changes in the file system.
- Explaining the key features of the .NET Framework's isolated storage mechanism.

Module 11: Internet Access

The following topics are covered in this module:

- Internet Application Scenarios
- The WebRequest and WebResponse Model
- Application Protocols
- Handling Errors
- Security
- Best Practices

After completing this module, you will be able to use the .NET Framework classes to work with data over the Internet. This includes:

- Using the basic request/response model to send and receive data over the Internet.
- Using the System.Net classes to communicate with other applications by using the Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Socket Internet protocols.

Module 12: Serialization

The following topics are covered in this module:

- Serialization Scenarios
- Serialization Attributes
- Object Graph
- Serialization Process
- Serialization Example
- Deserialization Example
- Custom Serialization
- Custom Serialization Example
- Security Issues

After completing this module, you will be able to serialize and deserialize an object graph. This includes:

- Writing an application that serializes and deserializes an object graph by using either a binary or Simple Object Access Protocol (SOAP) XML format.

Module 13: Remoting and XML Web Services

The following topics are covered in this module:

- Remoting
- Remoting Configuration Files
- XML Web Services

After completing this module, you will be able to create distributed applications by means of XML Web services and Object Remoting. This includes:

- Writing and configuring distributed applications that use .NET Remoting.
- Creating an XML Web service by using Visual Studio .NET and ASP.NET.
- Consuming an XML Web service by using the Web Services Description Language tool (Wsdll.exe).

Contact the training coordinator for pricing and details at 613-563-NOVA (6682) Ext:250 Or training@nova-networks.com

Nova Networks can also customize this course to topics of your choice which will reduce the course cost.

Copyright Statement

This site is Copyright © 2004 Nova Networks Inc. Reproduction of any part of this site for personal or commercial purposes without permission is strictly prohibited. The information at this site may be downloaded onto a disk or printed for your personal use provided that you include this copyright notice on each copy and that you make no alterations to any of the pages and do not use any of the information in any other work or publication whatsoever whether the publication is paper based or electronic. No part of the information may be distributed or copied for any commercial purpose.