



## Microsoft .NET

Total Duration: 182 Hours

### PRE-REQUISITE

Knowledge of computer programming and industry terms is required.

### .NET FRAMEWORK & DEVELOPING APPLICATIONS USING VB .NET

Duration: 49 hours

1. An Introduction to .NET
  - a. The .NET Framework
  - b. Languages and Web Support
  - c. Common Language Runtime
  - d. ADO.NET
  - e. .NET Enterprise Servers
  - f. Introduction to Visual Studio.Net
2. Introduction to Visual Basic
  - a. Need for Visual Basic .NET
  - b. Understanding .Net framework
  - c. Features of .NET
  - d. Features of VB.NET
  - e. The Architecture of VB.NET
  - f. System Requirements and Installation
  - g. Introduction to Visual Studio.NET
3. Introduction to Visual Studio.NET IDE
  - a. Introduction to Windows and Web Forms
  - b. Working with Visual Studio.NET
  - c. Creating a Project
  - d. Writing a simple application in Visual Basic.NET
4. Programming Concepts
  - a. Introduction
  - b. Declaring Variables
  - c. Types



- d. Enum Type
- e. Operators
- f. Statements in Visual Basic .Net
- g. Arrays
- 5. Procedures and Functions
  - a. Writing Procedures and Functions
  - b. Calling Procedures
  - c. Arguments
  - d. Passing Unknown Number of Arguments
  - e. Optional and Named Arguments
  - f. Scope of Variable
  - g. Useful Built-in Functions
  - h. Option Compare Statement
- 6. OOPs in Visual Basic.NET
  - a. Introduction to Object-Oriented Programming
  - b. Classes and Objects
  - c. Features of OOPs
  - d. Creating Classes
  - e. Constructors and Destructors
  - f. Creating the Properties, Methods and Events of a class
  - g. Namespaces
  - h. Imports Keyword
- 7. Inheritance, Polymorphism and Interfaces
  - a. Inheritance
  - b. Overriding Methods
  - c. MyBase Keyword
  - d. MyClass Keyword
  - e. Abstract Base Classes
  - f. Shared Members
  - g. Polymorphism
  - h. Interfaces
  - i. Structures
- 8. Exception Handling
  - a. Introduction to Exceptions
  - b. Unstructured Error Handling – On Error
  - c. Err Object
  - d. Structured Error Handling – Try Catch Finally Statement
  - e. Throwing Exceptions
  - f. User Defined Exceptions
- 9. Windows Forms
  - a. Base Classes of Windows Forms



- b. Visual Inheritance
  - c. Adding Controls
  - d. Simple Controls
  - e. Menus
  - f. Context Menus
10. Advanced Windows Application
- a. Advanced Controls
  - b. Writing MDI Applications
  - c. Graphical Applications
  - d. Custom Windows Control
  - e. Delegates

## C# .NET

Duration: 49 Hours

1. Introduction to the C# Language
  - a. The evolution of C#
    - i. Comparing different versions of C#
    - ii. Expressing C# models in UML
  - b. C# and the .NET infrastructure
    - i. Common Language Infrastructure (CLI)
    - ii. Managed code philosophy
    - iii. Intermediate Language (IL) and metadata
2. Language Fundamentals
  - a. Data types and control constructs
    - i. Declaring and initializing variables
    - ii. Value and reference types
    - iii. Unicode characters and strings
  - b. Defining and calling methods
    - i. The Main method specification
    - ii. Passing arguments and returning values
    - iii. The scope and lifetime of variables
    - iv. Static vs. instance methods
    - v. Handling exceptions
    - vi. Recovering resources
  - c. Employing .NET library classes
    - i. Avoiding collisions by using namespaces
    - ii. Performing input/output using the Console class and stream classes



- iii. Standard and Generic Collections
- 3. Developing C# Classes
  - a. Defining classes
    - i. Encapsulating attributes
    - ii. Writing properties with get and set
    - iii. Providing consistent initialization using constructors
    - iv. Overloading methods and constructors
    - v. Achieving reuse through inheritance and polymorphism
  - b. Creating and using objects
    - i. Allocating objects with new
    - ii. Passing initial values to constructors
    - iii. Choosing value or reference allocation
    - iv. Boxing and un-boxing
    - v. Invoking methods and accessing properties
- 4. Interconnecting Objects
  - a. Associating classes
    - i. Manipulating references
    - ii. Physical vs. logical equivalence
    - iii. Selecting collection library classes
    - iv. Increasing reliability using generics
  - b. Exposing interfaces
    - i. Defining an interface specification
    - ii. Implementing an interface in a class
    - iii. Interface polymorphism
    - iv. Indexers, events and delegates
    - v. Overloading operators
- 5. Simplifying Component Development
  - a. Component features of .NET
    - i. Manifests and assemblies
    - ii. Private vs. shared assemblies
    - iii. Deploying .NET components to the global assembly cache (GAC)
  - b. Writing .NET components in C#
    - i. Creating and calling custom components
    - ii. Extending System.ComponentModel.Component
  - c. Producing .NET components: a set of guidelines and standards
  - d. Interfacing to ActiveX components
    - i. Accessing COM/DCOM
    - ii. Tools for forward and backward compatibility
    - iii. Wrapping legacy components
  - e. Integrating Visual Basic 2005, C++ and C#
    - i. Harmonizing components through the CLI



- ii. Accessing metadata
- iii. Handling cross-language exceptions
- 6. Implementing and Enhancing C# Solutions
  - a. Building multitier applications
    - i. Generating user interfaces
    - ii. File I/O and serialization
    - iii. Accessing databases with ADO.NET 2.0
    - iv. Programming with DataSets
    - v. Distributing using Web services
  - b. Standards and versions
    - i. Standardization via ECMA/ISO
    - ii. Features in various C# standards
  - c. Advanced techniques
    - i. Automating documentation with XML
    - ii. Implementing operators, indexers and IEnumerator to support array-like behavior
    - iii. Attributes and reflection

## ASP .NET

Duration: 35 Hours

- 1. Introduction to ASP.Net
  - a. ASP.Net web Form Pages
  - b. Creating ASP.Net Pages, Basic Web Controls
  - c. Page Navigation, View State, Postback
  - d. Advance Controls (Validations, Calendar, AdRotator)
  - e. Runtime Programming with Controls
  - f. Creating Reusable Code
  - g. ADO.Net (Data Bindings, DataSet DataGrid)
  - h. Binding Data to a Repeater Control, Data list Control
  - i. Editable Data Grid
  - j. Editing record in a Data table
- 2. Introduction to ASP.net Applications
  - a. Using Global Application File
  - b. Configuring ASP.NET Applications
  - c. Overview of Web Config File
  - d. Handlers and Modules
  - e. Deploying ASP.NET Applications



- f. ASP.NET Sessions Web Forms State Management (Cookies, Session State, Session Events  
Cookie less Sessions)
  - g. Working with ASP.NET Applications (Caching, Page output Caching)
  - h. Page Fragment Caching, Page Data Caching
  - i. Application Errors, Page level Error Handling
  - j. Application Tracing
3. Security Applications
    - a. Introduction, Authentication Services, Forms Authentications
    - b. Windows Authentication
    - c. Configuring IIS, Configuring Windows Security
  4. Web Services
    - a. Introduction to Web Services, Architecture
    - b. Features and Benefits
    - c. Web Service Contracts
    - d. Creating a Web Services
  5. Consuming Client Web Applications
    - a. Windows Application and Web Applications
    - b. Data Driven Web Services

## ADVANCED FEATURES OF .NET 3.0/3.5

Duration: 49 Hours

### .Net 3.5 & VS 2008

#### Visual Studio 2008 and Language Features Overview

- Multi-targeting support
- C# and VB.NET language enhancements
- Language Integrated Query (LINQ)
- LINQ to SQL Designer

#### Building Web Applications with Visual Studio 2008

- New ASP.NET 3.5 controls
- ASP.NET AJAX integration and support for JavaScript IntelliSense
- JavaScript debugging.



## **Building WPF, WF and WCF Applications with Visual Studio 2008**

- Visual Studio 2008 WPF designer
- Building User Account Control (UAC) aware applications for Windows Vista
- Using the Windows WorkFlow designer
- Creating and consuming Windows Communication Foundation (WCF) services.

## **Visual Studio 2008 Office Development Features**

### **Module 1**

- .NET Framework 3.5 overview

### **Module 2**

- Query expressions
- Object initializers
- Local type inference
- Lambda expressions
- Extension methods
- Expression trees
- Anonymous types
- Partial methods

### **Module 3**

- Developing using LINQ
- Standard Query Operators
- LINQ to SQL
- LINQ to XML
- LINQ over DataSet
- LINQ to Entities

### **Module 4**

- HTML Designer
- CSS Support
- Javascript Support
- New Data Controls



- AJAX Support

#### **Module 5**

- Level-set features
- Visual IDE enhancements
- HTTP Programming Model
- JSON
- Services Syndication

#### **Module 6**

- Workflow Service Architecture
- WorkflowServiceHost
- WorkflowRuntime
- SharePoint Workflow template

#### **Module 7**

- WPF in .Net Framework 3.5 overview
- Building WPF solutions in VS 2008
- Integration with Expression Blend
- Data binding in WPF

#### **Module 8**

- Silverlight 1.0 plug-in overview
- Developing with Silverlight in Visual Studio 2008
- Silverlight and Javascript Intellisense

#### **Module 9**

- WPF using Silverlight
- XAML
- Using Shapes and Text
- Using Controls
- Canvas and Layout Managers
- Brushes
- Transforms
- Handling Events and Writing Code
- Building Custom UI Controls
- Reaching out and Programming the HTML of a page from a Silverlight control
- Handling HTML Events in Managed Code



- Exposing managed APIs to HTML JavaScript in the browser
- Using the File Open Dialog support
- Using the HTTP Network APIs
- Using the Web Service APIs
- Isolated Storage for local data caching

## **Module 10**

- WCF
- Architecture
- Service Orientation
- Web services specifications
- Programming model
- Services and endpoints
- Address, bindings, and contracts
- Channels and factories
- Serialization and versioning
- Hosting and activation
- JSON WCF Services in .Net 3.5

## **Module 11**

- Building/enabling a SOA application
- Business Process Execution Language (BPEL)
- What are Service Wrappers?
- What is WSDL?
- UDDI

## **Module 12**

- ASP.NET 3.5 Web Services
- Web Services Design
- Windows Communication Foundation
- Creating Web Services
- Configuring Protocols
- Web Service Bindings
- Document Vs Rpc
- Passing and Returning Types
- Binary Data
- Controlling XML Serialization
- WebMethod Attribute Parameters
- Validating messages



- Service Interfaces
- Soap Exceptions
- Soap Headers

## AT COURSE COMPLETION

At the end of this course participants will be able to develop and deploy secure Microsoft .NET client applications by using Microsoft Visual Studio® .NET and the Microsoft .NET Framework. At the end of this course participants will have an detailed information on implementing security; discusses design issues, including threat modeling techniques and coding techniques that enhance security; and explains why type-safety verification is the cornerstone of Microsoft .NET Framework security.