Windows Forms Programming Using Visual Basic.NET

Duration: 5 Days *(Face-to-Face & Remote-Live)*, or 35 Hours *(On-Demand)*

Price: CDN$3,275 *(Face-to-Face & Remote-Live)*, or CDN$1,995 *(On-Demand)*

Discounts: We offer multiple discount options. Click here for more info.

Delivery Options: Attend face-to-face in the classroom, remote-live or on-demand training.

Students Will Learn

- Using Visual Studio to create VB.NET applications

- Working with .NET data types

- Creating variables with the proper scope and using operators to build complex expressions

- Designing and using classes

- Using control structures such as `If`, `While` and `For`

- Using procedures to build complex applications

- Throwing and trapping exceptions using the `Try` and `Catch` statements

- Using single and multi-dimensional arrays

- Working with .NET collections

- Using LINQ to make queries

- Defining and implementing interfaces

- Working with enumerations

- Creating desktop-based applications

- Using control properties and methods to modify the appearance and behavior of controls

- Writing event handlers to respond to user interactions

- Using list-based controls such as `ListBoxes` and `ComboBoxes` on forms

- Using image controls such as `ImageList` and `PictureBox` on forms

- Accessing and displaying data using ADO.NET

- Binding data to controls on a Windows Form

- Using modal and modeless dialogs to interact with users

- Using the background worker to perform an asynchronous operation

- Calling Web services

- Reading and writing data from files and streams

- Deploying .NET applications

Course Description

This course provides students with hands on experience using Visual Studio to create desktop applications using Windows Forms and the .NET Framework using Visual

Basic.NET. The course provides a thorough introduction to the VB.NET programming language, including coverage of the essentials of the VB.NET programming language, built in data types, operators, control structures, classes and methods, collections and exception handling.

Students then learn how to leverage the power of the .NET Framework to build desktop applications. Students learn how to build Windows Forms applications and use with a variety of controls to create sophisticated user interfaces. Students also learn how to use the BackgroundWorker to perform asynchronous operations.

Students learn how to use ADO.NET to interact with databases and XML files. Students learn how Windows Forms uses data binding to display data in controls such as the DataGridView and Chart..Students will also learn how to leverage the power of web services by calling both SOAP services and RESTful APIs.

Other topics include: debugging techniques; using a .config file to control application configuration; building menus, toolbars and status bars; reading and writing files; interacting with the file system; and deploying desktop applications.

Comprehensive labs provide the students with extensive experience creating and deploying Windows Forms-based desktop applications.

This course provides thorough coverage of the use of **Windows Forms** to build desktop applications. Students requiring additional coverage of **Windows Presentation Foundation, ASP.NET Web Forms**, **ASP.NET MVC** or **WCF** should contact HOTT or refer to HOTT's complete course listing for additional training courses.

Students who are already familiar with the Visual Basic.NET language may want to take the 3-day Windows Forms Programming for Experienced VB.NET Programmers class instead.

Course Prerequisites

Prior programming experience is required.

Course Overview

Introduction to .NET

- Overview of the .NET Framework
- How .NET is Different from Traditional Programming
- Common Language Runtime (CLR)
- Common Language Specification (CLS)
- Common Type System (CTS)
- .NET Assemblies
- Microsoft Intermediate Language (CIL)
- .NET Namespaces
- .NET Framework Class Library

Introduction to Visual Studio

- Creating a Project
- Using the Code Editor
- Correcting Syntax Errors
- Setting Project Properties
- Adding References
- Compiling a Program
- Running a Program
- Debugging a Program
- Using the MSDN (Help)

Language Fundamentals

- VB.NET Program Structure

Conditionals and Looping

- If/Else

- Defining Namespaces
- Understanding VB.NET Data Types
- Defining Variables and Constants
- Comparing Value Types vs. Reference Types
- Working with Operators and Expressions
- Performing Type Conversions
- Using Console I/O
- Formatting Numbers, Date and Times

- `Select Case`
- `Do/Loop`
- `While`
- `For`
- `For Each`

Procedures and Parameters

- Subroutines vs. Functions
- Defining Shared and Instance Methods
- Passing Parameters by value and by reference
- Overloading Methods
- Using Variable Length Parameter Lists

Exception Handling

- What are Exceptions?
- .NET Exception Hierarchy
- Catching Exceptions
- Throwing Exceptions
- Managing Resources with Finally

Collections

- Defining and Using Arrays
- Understanding `System.Array`
- .NET Collections vs Generic Collections
- Working with Lists
- Working with Dictionaries
- Using LINQ to Objects

Object-Oriented Programming

- Overview of Object-Oriented Programming
- Building Classes
- Defining Properties
- Using Auto-Implemented Properties
- Defining Methods
- Understanding Constructors
- Extending .NET Classes via Inheritance
- Defining and Implementing Interfaces
- Understanding the Role of Interfaces in .NET

Windows Forms Applications

- Windows Forms Applications
- Setting Form Properties
- Understanding the Life-cycle of a Form
- Using the Windows Forms Designer
- Using the `MessageBox` Class
- Using a `.config` File

Using Controls

- Working with Windows Forms Controls
- Using `Text` Controls
- Using `Button` Controls
- Using `Selection` Controls
- Using `List` Controls
- Using `Container` Controls
- Using `Image` Controls
- Using `Up/Down` Controls
- Using the `ErrorProvider` and `ToolTipProvider` Controls

Handling Events

- Understanding the Event-Driven Programming Model
- Writing Event Handlers
- Sharing Event Handlers

Performing Asynchronous Activities

- Understanding Threading
- Working with Delegates
- Using the Background Worker
- Updating Controls from Other Threads

ADO.NET

Using XML

- Understanding the ADO.NET Object Model
- Connected vs. Disconnected Access
- Using a Connection to Connect to a Data Source
- Using a Command to Execute Queries and Stored Procedures
- Using a `DataReader` to Work with Cursors
- Using the `DataSet` with Disconnected Data
- Using `DataAdapters` with DataSets

- Understanding XML and XML Schemas
- Reading XML Data with a `DataSet`
- Writing XML Data with a `DataSet`

Data Binding

- Understanding ADO.NET Data Binding
- Binding to Simple and Complex Controls
- Manually Binding Controls
- Using the `BindingSource` Control
- Using the `BindingNavigator` Control
- Using the `DataGridView` Control
- Using the `Chart` Control

Working with Forms

- Understanding Modal vs Modeless Forms
- Displaying Modal Forms
- Working with `DialogResult`
- Retrieving Data from Modal Forms
- Displaying Modeless Forms
- Working with Data in Modeless Forms
- Using the Common Dialogs

Working with Menus, Toolbars and Status Bars

- Working with Menus
- Working with Toolbars
- Working with Status Bars

Working with Files and Directories

- Working with the Windows File System
- Discovering Drives
- Discovering Directories
- Discovering Files
- Examining File, Directory and Drive Attributes
- Reading, Writing and Appending to Files

Calling Web Services

- Understanding Web Services
- Calling SOAP Services
- Calling REST Services
- Serializing and Deserializing JSON

Deploying Windows Forms Applications

- Understanding Deployment Options
- Configuring an Application for Deployment
- Using XCOPY Deployment
- Using Installers
- Using Click-Once Deployment