Course Outline | VB.NET Introduction & Intermediate

5 day(s)

Overview

This course starts with a quick overview of the .NET platform, examining assemblies, Microsoft Intermediate Language, Visual Studio profiles, XML comments, IntelliSense, and debugging. From there, you will learn all the Visual Basic language features that you must understand in order to create full-featured Web or Windows applications that make the best use of the .NET platform.

You will learn about data types, variables, and operators, along with all the important flow control structures. You'll work through several examples demonstrating the power of the .NET Framework, and dig into creating and consuming your own classes and objects. The course moves on to working with data structures, such as arrays and collection classes, before finishing up with discussions of generics, handling exceptions and working with delegates and events. By the end of this course, you will understand the important basic concepts that will allow you to start creating .NET applications.

Prerequisites

Some prior programming experience is useful for this course, but not essential.

Course Outline

Getting Started with .NET

- Thinking about .NET
- Using Visual Studio 2008
- Debugging Your Code and Handling Exceptions
- Generics
- Core .NET Language Techniques
- Object-Oriented Techniques

Data Types and Variables

- Introducing Variables and Data Types
- Working with Variables and Data Types

Using the .NET Framework

- Using the .NET Framework Classes
- Working with Strings
- Working with Dates and Times
- The My Namespace

Branching and Flow Control

- Conditional Branching
- Repeating Code Blocks
- Unconditional Branching

Classes and Objects

- Introducing Classes and Objects
- Working with Classes

Properties and Methods

- Working with Properties
- Working with Methods

Working with Arrays

- Introducing Arrays

- Manipulating Arrays

Generics

- Introduction Generics
- Generics and Arrays
- Generic Interfaces
- Generic Constraints
- Generics and Lists

Handling Exceptions

- Perspectives and Exception Handling
- Getting Started with Exception Handling
- Catching Specific Exceptions
- Raising Errors
- Running Code Unconditionally
- Creating Exception Classes

Collection Classes

- Generics, Collections, and Interfaces
- The Generic List
- Working with Dictionaries, Stacks, and Queues
- Creating Your Own Generic Collection Classes

Core .NET Language Techniques

- The Role of Implicitly Typed Local Variables
- The Role of Extension Methods
- Object Initialization Syntax
- The Role of Anonymous Types
- The Role of Lambda Expressions

Object-Oriented Techniques

- Creating Your Own Classes
- Inheritance
- Interfaces
- Organizing Classes

Delegates and Events

- Motivating Delegates
- Introducing Delegates
- Working with Events