

The VB.NET Programming Language

Duration: 5 Days · **Training format:** Virtual Classroom / Online

Description

Microsoft's .NET Framework presents developers with unprecedented opportunities. From web applications to desktop and mobile platform applications - all can be built with equal ease, using substantially the same skill-set. But to make the most of this potential, developers must have a thorough grasp of core language skills and OO programming concepts.

It concentrates on the programming language itself, to prepare delegates fully in readiness for exploring the .NET Framework. No Object-Orientated knowledge is assumed - the course provides a suitable OO primer. From basic procedural syntax to sophisticated object-oriented programming techniques, delegates will learn how to write .NET applications with code that is robust and maintainable.

The course is presented as a mixture of lectures, demos and hands-on exercises. Practical sessions follow all main topics, designed to reinforce the points covered. Additional information is provided in appendices to extend the learning experience after the course has been completed.

Delegates will learn how to

- Write code that includes sequence, selection and iteration constructs
- Create and use classes and structures (types), including fields, properties and methods
- Use Private, Friend, Protected and Public visibility modifiers
- Use exception-handling to create robust applications
- Create derived classes that inherit from custom-written or .NET Framework classes
- Create interfaces and apply techniques of polymorphism effectively and appropriately
- Work with generic types
- Leverage the power of VB.NET features
- Use delegates
- Understand the event handling paradigm
- Use lambda expressions in code
- Use LINQ (Language Integrated Query)
- Efficiently manage resources

Outline

Introduction to Object Orientation

OO and VB.NET

Introduction to .NET, Visual Studio & VB.NET

- The .NET Framework; The Common Language Runtime; The Common Type System
- .NET Features; Introduction to namespaces and assemblies
- Get to know your way around Visual Studio

Syntax I

- Procedures and statements; Data types; Declaring variables; Assignments
- Conversion; Arithmetic and other operators
- Flow of control constructs
- Passing parameters by value, by reference, named and optional parameters

Syntax 2

- Type concepts; Classes; Reference types
- Fields, properties and methods
- Accessibility modifiers
- Object initialisation, Constructors and Constructor chaining
- Instance members; Keyword 'Me'
- Enumerated types
- Arrays

Collections

- Generic Collections
- Iterating collections
- Indexers

Inheritance & Polymorphism

- Concept of inheritance; Substitutability; Extending a simple class
- 'Overridable', 'Overrides' and 'NotOverridable' modifiers
- Polymorphism
- Abstract classes (MustInherit)
- Upcasting and safe downcasting

Interfaces

- Polymorphism with interfaces
- Multiple interfaces

Delegates and Lambdas

- Delegates explained
- Working with delegates
- Creating your own delegate types
- The evolution of Lambdas

Generic Delegates

- Using the framework-supplied generic delegates

LINQ

- The language features behind LINQ
- The LINQ API and query pattern
- Grouping and Joins

Handling Exceptions

- Errors vs. Exceptions
- The 'try', 'catch', 'finally' paradigm
- Using 'throw'
- Creating your own exceptions

Consuming Events

- Understanding events in .NET
- Writing event handlers

Producing Events

- The event conventions
- Raising custom events

Garbage Collection

- Garbage collection and its impact
- Finalizers; The 'Dispose' pattern; IDisposable
- The 'using' statement

The Way Ahead

- Review
- Follow-on courses

Prerequisites

There is no prerequisite