Learning to Program with VB.NET

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

Price: $2495 (Face-to-Face & Remote-Live), or $1495 (On-Demand)

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

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 the .NET data types
- Creating variables with the proper scope and using operators to build complex expressions
- Using control structures such as If, Do While and For
- Using procedures to build complex applications
- Designing and using classes
- Using arrays and .NET collections
- Throwing and trapping exceptions using the VB.NET Try and Catch statements
- Using ADO.NET classes to access databases
- Working with files and directories
- Building simple Windows Forms applications
- Building and using derived classes
- Defining and implementing interfaces
- Performing advanced string manipulation

Course Description

This hands on VB.NET programming course provides an introduction to programming using the VB.NET language. Students are introduced to the application development cycle, structure of programs, and specific language syntax. The course also contains "Thinking Like a Programmer" sections that provide students insight on how to develop common algorithms. The course covers console and file I/O, string and character manipulation, managing data using collections and fundamental object-oriented programming concepts. Error handling techniques are also emphasized. The course also introduces how to access databases using ADO.NET and illustrates how to build user interfaces using Windows Forms. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

This class is intended for non-programmers. Students who already understand fundamental structured programming and object-oriented techniques should attend either ASP.NET Web Forms Programming Using Visual Basic.NET, Windows
Course Prerequisites

Familiarity with computers.

Course Overview

<table>
<thead>
<tr>
<th>Application Development Fundamentals</th>
<th>Managing the Flow of an Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of Programming Concepts</td>
<td>Conditional Constructs</td>
</tr>
<tr>
<td>Understanding the Structure of a VB.NET Program</td>
<td>Working with If/Else Constructs</td>
</tr>
<tr>
<td>Understanding Data Types</td>
<td>Working with Select Case Constructs</td>
</tr>
<tr>
<td>Working with Variables</td>
<td>Working with Do/Loop Constructs</td>
</tr>
<tr>
<td>Reading From and Writing to the Console</td>
<td>Working with For and For Each Constructs</td>
</tr>
<tr>
<td>Overview of the .NET Framework</td>
<td></td>
</tr>
<tr>
<td>Using Visual Studio</td>
<td>Thinking Like a Programmer</td>
</tr>
<tr>
<td>Thinking Like a Programmer</td>
<td>Making Decisions</td>
</tr>
<tr>
<td>Designing Algorithms</td>
<td>Designing Loops</td>
</tr>
<tr>
<td>Finding Patterns in Code</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Working with Data</th>
<th>Using Procedures to Modularize Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using Integer Data Types</td>
<td>Defining and Calling Subroutines</td>
</tr>
<tr>
<td>Using Floating Point Data Types</td>
<td>Defining and Calling Functions</td>
</tr>
<tr>
<td>Using Characters and Strings</td>
<td>Understanding Variable Scope</td>
</tr>
<tr>
<td>Using Dates</td>
<td>Overloading Procedures</td>
</tr>
<tr>
<td>Using Booleans</td>
<td>Passing Parameters</td>
</tr>
<tr>
<td>Working with Constants and Literals</td>
<td>Understanding ByVal vs ByRef Parameters</td>
</tr>
<tr>
<td>Understanding Option Strict</td>
<td>Understanding the Call Stack</td>
</tr>
<tr>
<td></td>
<td>Thinking Like a Programmer</td>
</tr>
<tr>
<td></td>
<td>Finding the Procedures</td>
</tr>
<tr>
<td></td>
<td>Refactoring</td>
</tr>
<tr>
<td></td>
<td>Improving Productivity with Snippets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object-Oriented Programming</th>
<th>Understanding Namespaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding Object-Oriented Concepts</td>
<td>Understanding the Role of Namespaces</td>
</tr>
<tr>
<td>Designing Classes</td>
<td>Understanding .NET Namespaces</td>
</tr>
<tr>
<td>Coding Properties and Methods</td>
<td>Defining Custom Namespaces</td>
</tr>
<tr>
<td>Initializing Objects with Constructors</td>
<td>Referencing Members in a Namespace</td>
</tr>
<tr>
<td>Overloading Constructors</td>
<td>Using the Imports Statement</td>
</tr>
<tr>
<td>Declaring and Instantiating Objects</td>
<td></td>
</tr>
<tr>
<td>Calling Properties and Methods</td>
<td></td>
</tr>
<tr>
<td>Understanding Value Types vs. Reference Types</td>
<td></td>
</tr>
<tr>
<td>Working with Shared Data Members</td>
<td></td>
</tr>
</tbody>
</table>
### Working with Data Collections
- Understanding Arrays
- Declaring and Instantiating Arrays
- Iterating through Arrays
- Working with System.Array Methods
  - Copying, Sorting, Searching and Resizing
- Passing Arrays to Methods
- Working with ParamArray Parameters
- Working with Command-line Arguments
- Understanding .NET Collections
- Managing Data Using `List(Of T)`
- Managing Data Using `Dictionary(Of TKey, TValue)`
- Working with LINQ
- Making LINQ Queries
- Enumerating LINQ Query Results
- Working with Anonymous Types
- Using Extension Methods with LINQ

### Building Inheritance Hierarchies
- Understanding Inheritance
- Building Derived Classes
- Understanding Constructors in Derived Classes
- Defining and Using Protected Class Members
- Understanding Polymorphism
- Defining Overridable Methods
- Overriding Methods
- Understanding Abstract Classes
- Defining Abstract Classes
- Inheriting from Abstract Classes

### Building GUIs with Windows Forms
- Overview of Windows Forms
- Designing Forms
- Working with Controls
  - Using Labels and Textboxes
  - Using Buttons
  - Using Checkboxes and Radio Buttons
  - Using Menus
  - Using list Controls
- Handling Events

### Accessing Databases Using ADO.NET
- Understanding the ADO.NET Object Model
- Opening Connections
- Executing Queries Using Commands
- Iterating Through Results Using DataReaders
- Calling Stored Procedures
- Passing Parameters to Stored Procedures
- Working with DataSets
- Binding Data to Controls

### Exception Handling
- Understanding Exception Handling
- Using Try/Catch to Handle Exceptions
- Working with the Exception Class
- Understanding Exception Propagation
- Using Finally to Manage Cleanup Processing
- Throwing Exceptions

### Working with Files, Directories and Streams
- Using the System.IO Namespace
- Discovering Drives
- Working with Directories
- Working with Files
- Parsing a File Path
- Understanding Streams
- Working with FileStream
- Reading and Writing Text Files
- Understanding other Types of Streams

### Working with Strings

### Building N-Tier Applications
Working with the String Class
- Working with String Literals and Escape Sequences
- Understanding String Manipulation Performance Issues
- Working with the StringBuilder Class
- Formatting Output with String.Format

Building Large Scale Applications
- Designing N-Tier Applications
- Building .NET Assemblies
- Referencing Assemblies

Hands On Technology Transfer
The Best Way to Transfer Technology Skills

1 Village Square, Suite 8
14 Fletcher Street
Chelmsford, MA 01824

Copyright © 2020