

Mastering VB.NET using Visual Studio 2010

Course Length: 5 days

Price: \$2,500

Summary

Each day there will be a combination of presentations, code walk-throughs, and hands-on projects. The final project (Friday) incorporates everything we've covered during the week including some new features from the .NET 4.0 Framework

Who Should Attend?

Students should be familiar with VB.NET programming syntax and the .NET Framework base classes.

Course Outline

NET Essentials

- New Component Model
- Common Language Runtime
- Base Class Library
- Common Type System
- Namespaces and Classes
- Code Access Security
- .NET Compile Process
- Assemblies
- Intermediate Language
- Assembly Metadata
- Decompiling with ILDASM
- Protecting Intellectual Property
- Obfuscation Tools

- Using Reflector
- Sharing Assemblies
- Disk and Memory Usage

Visual Studio IDE

- Overview of the new IDE for VS2010
- IDE Enhancements: Edit and Continue, Object Debugging, Intellisense Filtering and Autocorrect, Snippets, XML Documentation and more
- Avoid Docking Hell
- Using the IDE Effectively
- Solutions and Projects
- Solution Explorer
- Toolbox
- Server Explorer
- Class View
- Property Window
- Immediate Window
- Autos and Locals windows
- Output Window
- IDE and Debug Options

Working with Code

- Using Code Regions
- Windows Forms Event Handling
- Handles Clause
- AddHandler and RemoveHandler
- Controls vs Components

- InitializeComponent and the Designer
- Dynamic Help
- Debugging
- Edit and Continue

Objects Part 1

- Value Types vs Reference Types
- ByVal vs ByRef
- Structures vs Classes
- Performance Considerations
- Scope Keywords
- Constructors
- Overloaded Members
- Finalizers and Dispose

Objects Part 2

- Inheritance
- Casting
- Overriding
- Shadowing
- Interfaces and Polymorphism
- OOP in the Real World
- Customizing Windows Forms Controls
- Object Binding
- Implementing Plug-Ins with Interfaces and Reflection

Strings, Arrays, and Collections

- Generics
- String Class
- StringBuilder Class
- Object Arrays
- ArrayList Class
- Performance Considerations
- HashTable Class
- Custom Collection Classes
- Binding Arrays to Windows Forms Controls

Streams and Serialization

- System.IO Namespace
- Stream Class
- StreamReader and StreamWriter Classes
- FileStream Class
- MemoryStream Class
- Compression Streams
- Encryption Streams
- NetworkStream Class
- Serializing and Deserializing Objects

Auto-Deployment of Windows Apps

- Windows' Painful Deployment History
- .NET Spells Relief
- Security Issues
- Strategies for Auto-Deployment

- Trusted Sites List Solution
- NetRun Solution
- .NET Loader Solution
- Strongly-Named Assembly Solution
- Setting up and working with Click-Once Deployment

Data and Access

- ADO.NET Breakdown
- DataSets Overview
- Using Datasets without a database!
- Reading and Writing XML
- Methods of Accessing Data
- Using Command Objects
- Returning a DataReader
- Returning a single value
- Performing an insert or update query
- Drag and Drop Stored Procedure calls!
- DataAdapters Overview
- Maximizing Efficiency
- How .NET manages concurrency
- Keeping Datasets and Databases in sync
- Limitations of DataAdapters
- Using Typed DataSets
- Pros and Cons of Typed Datasets
- Creating and using XSD Schemas

- How ADO.NET uses XML
- Why .NET Binding is better than VB6 Binding
- Using DataViews to sort and filter
- Calling Stored Procedures manually
- Returning hierarchical data from multiple tables into a Dataset
- Using transactions with DataAdapters
- Connection String irregularities
- Business Objects: Enforcing Rules
- Enforcing Rules in Real Time in the business object!
- Working with DataSources for prototyping (deeper coverage in ASP section)
- Dataset enhancements in 2010
- Working with SQL Server 2010 XML datatypes
- Using SQL BulkCopy
- Using Provider Factories
- Serialization Enhancements
- Best Practices

Web Services: Why Web Services?

- Web Service Support in the Industry
- Protocols: UDDI, DISCO, WSDL, and SOAP
- Three steps to a web service
- Accessing a web service in any .NET App
- Exposing existing COM and .NET Objects
- Passing Datasets through Web Services
- Best Practices for designing Web Services

- Securing Web Services
- Best Practices

Asynchronous Calls (Threads)

- Why use Asynchronous Calls?
- Calling Web Services Asynchronously
- Creating new threads
- Thread Synchronization
- Race Conditions
- Inter-thread communication
- Executing multiple tasks in parallel
- Accessing Windows UI from other threads.
- Using Delegates

Windows Services

- Windows Services Overview
- Building a Service
- Installers
- Installing the service
- Running with Service Manager
- Best Practices

Console Applications

- Console overview
- Writing console apps
- Scripting!

Instrumenting your code

- Reading and Writing Performance Counters
- Writing to Event Logs
- Tracing
- Best Practices

Deployment, Versioning and Configuration

- The problems with COM
- Assemblies undercover
- Assembly Manifest
- Private vs. Shared .NET components
- Strong Names
- Deploying to the GAC
- Assembly binding and redirection
- Deploying Windows apps with the browser
- Using the Codebase parameter
- Dynamic Properties
- Persisting Connection Strings
- Best Practices

Encryption

- Encryption Basics
- RSA vs. DES Encryption
- String/File Encryption
- Encrypting Datasets
- Best Practices

Introduction to Windows Workflow Foundation (WF)

- Building Simple Workflows
- Intrinsic Workflow Activities
- Building Custom Activities
- Using Workflow in Windows Forms Applications
- Using Workflow with ASP .NET Applications

Introduction to Windows Communication Foundation (WCF)

- Enterprise Services, Remoting, or WCF
- WS-* Protocols
- Building a WCF Service
- Building a WCF Consumer
- Working with Data Contracts

Introduction to Windows Presentation Foundation (WPF) Raster and Vector Graphics

- WPF Configuration
- XAML
- Integration with Visual Studio
- Creating a WPF Application