# Course 462: VB.NET Programming and the .NET Framework (5 days)

## Course Description…

This 5-day program is a beginning to intermediate level course covering the major features of the VB.NET language. Topics introduced include VB.NET syntax, object-oriented principles, and the .NET Framework.

Through a combination of workshops, lecture and discussion, the principles of encapsulation, inheritance (both implementation and interface), and polymorphism are introduced. Event handling in VB.NET is covered in detail, and delegate classes are introduced.

Attendees will learn how to create both private and shared assemblies, and will be introduced to such concepts as memory management, thread safety, and exception handling. A brief overview of ADO.NET and ASP.NET web applications and web services is also provided.

Workshops and hands-on exercises constitute almost 60% of this course and students will construct a working application that demonstrates good coding practices for writing solid, event-driven, object-oriented code that can be deployed as private or shared assemblies and across a network.

## Learning Objectives…

- Understand the syntax of the VB.NET programming language.
- Briefly review procedural syntax: variable declaration, functions, conditional branches and loops.
- Understand implementation and interface inheritance and when they are used
- Discover the power of loosely coupled events, how to declare them in classes and use them effectively in client-side code
- Explore the significance of polymorphism and its role in supporting robust and well-formed classes
- Briefly introduce the concept of components (private and shared assemblies)
- Briefly introduce the .NET strategy for designing web applications and web services
- Briefly introduce the .NET data access model

## Who should attend…

Audience includes individuals who have some familiarity with programming in any language.

## Prerequisites…

Some prior programming experience is recommended.

# Course Outline…

**Introduction and Overview**

Course Objectives

## Unit 1: Introduction to .NET

**What is .NET?**

- .NET Framework
- Microsoft Intermediate Language
- Common Language Runtime

**Why is .NET?**

- Safer, more secure and more stable
- Run once, run always
- XCOPY installation and the elimination of DLL Hell
- Multi-language development within a single enterprise

## Unit 2: Introduction to Visual Studio .NET

**The Visual Design Process**

- Draw interface
- Set design-time properties
- Write code

**Tour of the Rapid Application Development (RAD) Environment**

- Menus and toolbar buttons
- Windows
- Files
- Project types

## Unit 3: Introduction to GUI Programming

**Tour of Basic Visual Components**

- Event-driven programming and the GUI
- The Main Method
- Forms
- Simple controls
- Simple debugging tools

**Tour of Advanced Visual Components**

- Menus
- Dialogs
- Advanced Controls

**Program Distribution**

- The need for the .NET Framework
- XCOPY Distribution
- Package and Deployment Wizard

# Unit 4: Introduction to Programming Syntax

**Statements**

- Variables
- Constants
- Operators
- Debugging using
  - breakpoints
  - single-stepping
  - the immediate window

**Program Control**

- Sequence
- Conditional Branching
- Conditional Looping
- Debugging using the Watch List

**Program Modularity**

- Procedures, parameters and method overloading
- Variable scope
- Debugging using the Call Stack

# Unit 5: Introduction to Object-Oriented Programming

**The Three Elements of an Object-Oriented Program**

- Encapsulation
- Inheritance
- Polymorphism

**The Reasons Behind Object-Oriented Programming**

- Reusability
- Maintenance
- Clarity

# Unit 6: Encapsulation

**Principles**

- Code and data together
- Data hidden behind code
- Interface and implementation as separate entities

**Classes**

- What is a class? (classes as blueprints, objects as homes)
- Building a class (fields, methods, property procedures)
- Static variables, methods and classes
- Property procedures
- Namespaces

**Objects**

- Declaring a class (reference types vs. value types, declaration syntax)
- The difference between classes and structures (boxing and unboxing)
- Potential for memory leaks with reference types
- The .NET Garbage Collector

**Other Topics**

- Constructors (including overloading)
- Visual Studio tools for debugging classes
- The Object Browser

## Unit 7: Inheritance and Polymorphism

**Implementation Inheritance**

- Inheriting from a single parent
- Extending a parent
- Overriding a parent
- Calling a parent's constructor
- Abstract classes
- Abstract methods
- Visual inheritance (forms and controls)
- Cross-language inheritance and debugging
- The Object Browser revisited

**Interface Inheritance**

- Inheriting interfaces from many parents
- Interface as a contract
- Distinguishing implementation and interface inheritance
- Examples

**Polymorphism**

- Polymorphism as a substitute for some conditional branches
- Polymorphism via both implementation and interface inheritance
- Polymorphism as the vehicle for robust, reusable classes
- Examples

**The Collection Namespace**

- Collection objects (hashtable, arraylist, stack, queue, etc)
- Debugging with objects and collections of objects
- Examples (using polymorphism as a key example)

# Unit 8: Events

**Loose Coupling of Events**

- Binding an object's events to an event handler
- Defining events within a class
- Examples

**The Standard Event Signature**

- The role of "sender as object" in the standard event signature
- The role of "e as eventargs" in the standard event signature
- Defining events within a class using the standard event signature
- Using the standard event signature in an event handler
- Examples

# Unit 9: .NET Framework Programming

**Thread-safe applications**

- Introduction to preemptive-scheduling
- The Thread class
- SyncLock

**Delegates**

- The role of delegates in events
- The role of delegates in multi-threaded applications
- A general role for delegates

**Components**

- Private assemblies
- Shared assemblies
- Review of namespaces
- Legacy components (COM components and VB6 conversion)

# Unit 10: Data Access

- File I/O
- Where we are coming from (ODBC and OLE DB)
- The need for the managed provider
- Comparing ADO with ADO.NET
- ADO and ADO.NET Connection and Command objects
- The evolution of the Recordset Object
- DataReader objects
- DataAdapter and Dataset objects
- Dataset objects and XML
- Dataset objects and the IEnumerable interface

## Unit 11: ASP.NET Web Applications

- Introduction to the Web
- ASP.NET and its object-oriented, event-driven approach
- Visual Studio and the code-behind model
- HTML and WebForm controls
- The Page Object
- Managing State

## Unit 12: ASP.NET Web Services

- Marshalling
- Building  a Simple Web Service
- XML, SOAP, and WSDL
- Serializing Objects in a Web Service
- Parsing XML in a Web Service
- Asynchronous Web Services

## Unit 13: Putting It All Together

**What Has Been Learned**

- Understand implementation and interface inheritance and when used
- Discover the power of loosely coupled events, how to declare them in classes and use them effectively in client side code
- Explore the significance of polymorphism and its role in supporting robust and well-formed classes
- Show the syntactical rules of one of the .NET languages (either VB.NET or C#)
- Briefly introduce the concept of components (private and shared assemblies)
- Briefly introduce the .NET data access model

***Please contact your ROI representative to discuss course tailoring!!!***