

Learning to Program in Visual Basic 2005

Course Description: In this course, you'll learn to use Visual Studio 2005 to explore the Visual Basic 2005 language. The 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'll learn all the language features that you must internalize in order to create full-featured Web or Windows applications that make best use of the .NET platform. You'll 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 the applications you need.

Prerequisites: Some experience with programming is recommended.

Course Contents

Getting Started with .NET

- Thinking about .NET
- What is .NET?
- Why .NET?
- Advantages of .NET
- .NET Architecture
- Examining a .NET Application:
- Introducing Assemblies
- What Kinds of Programs Can You Create?
- Using the Command-Line Compiler to Build a Console Application
- Revisiting Microsoft Intermediate Language
- Using Visual Studio 2005
- Working with Your Visual Studio Profile Settings
- Revisiting the .NET Console Application
- Debugging and Handling Exceptions
- Handling Errors
- Fixing the Logic Error

Data Types and Variables

- Introducing Variables and Data Types
- Variables
- Data Types
- Integer Data Types
- Floating-Point Data Types
- Decimal Data Type
- Char Data Type
- String Data Type

Data Types and Variables (contd)

- Boolean Data Type
- Date Data Type

- Object Data Type
- Constants, Enumerations, and Structures
- Working with Variables and Data Types
- Converting from One Data Type to Another
- Operators

Using the .NET Framework

- Using .NET Framework Classes
- Generating Random Numbers
- Getting Information about the Computer
- Working with XML
- File Input/Output
- Working with Strings
- Fields and Properties of the String Class
- Methods of the String Class
- Formatting Strings
- Using the StringBuilder Class
- Working with Dates and Times
- Properties of the DateTime Structure
- Methods of the DateTime Structure
- Using the TimeSpan Structure
- The My Namespace
- My.Application
- My.Computer
- My.User



CLEAR CANYON[®]
TECHNOLOGIES_™
It's smart to learn from the best!

Branching and Flow Control

- Branching in Code
- Conditional Branching
- Repeating Code Blocks
- Unbounded Looping
- Looping for a Specific Number of Times
- The For Each Loop
- Unconditional Branching
- Exit For/Exit While/Exit Do
- The Goto Statement
- The Continue Statement

Classes and Objects

- Introducing Objects and Classes
- Revisiting .NET Framework Classes
- Using Class Properties
- Calling Class Methods
- Creating Your Own Classes
- Creating Your First Class
- Adding XML Comments
- Using the Class View Window
- Using the Class Designer
- Investigating the Code
- Adding the Code
- Add another Method and Test the Class
- Working with the Object Test Bench
- Working with Classes
- Value Types vs. Reference Types
- Setting Reference Variables to Nothing and Disposing
- Providing for Deterministic Finalization
- Overriding Object Class Functionality
- Manipulating Object References
- Instance vs. Shared members
- Thinking More about Shared Members

Properties and Methods

- Overview of Properties and Methods
- Properties
- Calculating Values for Properties
- Validating Values in Properties
- Passing Arguments to Properties
- Methods
- Passing Arguments to Methods
- Methods in Classes
- Class Constructors
- Saving and Retrieving Information
- Returning and Passing Arrays
- Optional Parameters
- Named Parameters

Instance and Static Members Object-Oriented Techniques

- Inheritance
- Derived Classes
- Abstract Classes and Members
- Sealed Classes and Members
- Interfaces
- Implementing an Interface
- Interfaces in the .NET Framework
- Organizing Classes
- Partial Classes
- Nested Classes
- Namespaces

Working with Arrays

- Introducing Arrays
- Creating and Filling Arrays
- Initializing Arrays
- Retrieving Data from Arrays
- Arrays and Method Parameters
- Arrays in the .NET Framework
- Manipulating Arrays
- Sorting Arrays
- Allowing for More Flexible Sorting
- Searching in Arrays

Delegates and Events

- Motivating Delegates
- Testing the FileSearch0Class
- FileSearch1: Modifying the FileSearch0 Class
- FileSearch1a: Inheritance is a Possible Solution
- Using an Event Interface Class
- Introducing Delegates
- Digging Deeper into Delegates
- Delegate vs. MulticastDelegate
- Working with Events
- Doing Things the VB6 Way
- Multiple Event Handlers
- Exceptions and Multiple Event Handlers
- Manually Invoking Each Listener
- Using the .NET Event Design Pattern
- Adding and Removing Handlers Dynamically



CLEAR CANYON[®]
TECHNOLOGIES
It's smart to learn from the best!

Generics

- Introducing Generics
- Generic Methods
- Generic Classes
- Advantages of Generics
- Generics and Arrays
- Sorting Arrays
- Sorting with the IComparer Interface
- Sorting with Generic Comparisons
- Searching with Generic Predicates
- Generic Interfaces
- Generic Constraints
- Generics and Lists
- Using an ArrayList to Store a Collection of Items
- Using a Generic List to Store a Collection of Items
- Sorting with the List Class

Handling Exceptions

- Perspectives on Exception Handling
- Exception Handling in .NET
- The Demonstration Code
- Getting Started with Exception handling
- No Error Handling
- Adding a Simple Try/Catch Block
- Unhandled Exceptions
- Using an Exception Object
- Catching Specific Exceptions
- Understanding the Exception Hierarchy
- Raising Errors
- Using the Throw Keyword
- Exception-Handling Options
- Passing Error Information
- Running Code Unconditionally
- The Finally Block

Collection Classes

- Generics, Collections, and Interfaces
- Introducing the Generic Collection
- Interfaces
- The Generic List
- Working with the List Class
- Working with Predicates
- Working with Dictionaries, Stacks, and Queues
- Storing Key/Value Pairs Using a Dictionary
- Using the SortedDictionary and SortedList Classes
- Using Queues and Stacks
- Creating Your Own Generic Collection Classes
- Creating the Custom Collection Class
- Consuming the Custom Collection