

This two-day intensive course is designed for the experienced programmer to help you quickly come up to speed on the C# language. It is current to Visual Studio 2010 and .NET 4.0, which introduces important new features such as dynamic data type, named and optional arguments, and the use of variance in generic interfaces. The new features are covered in a new chapter. A new appendix covers the fundamentals of Language Integrated Query (LINQ), which was introduced with .NET 3.5. This course concisely covers the essentials of programming using Microsoft's C# programming language. It starts with a brief chapter, ".NET: What You Need to Know," which gets you up and running in the .NET environment with a minimum of fuss. The next two chapters cover C# language essentials and object-oriented programming in C#. The next chapter discusses how C# relates to the .NET Framework. The following chapter covers delegates and events. The course includes a succinct introduction to creating GUI programs using Windows Forms. The course concludes with a chapter covering the new features in C#. Appendices provide a tutorial on Visual Studio 2010 and an overview of LINQ. The course is practical, with many example programs and a progressively developed case study. The goal is to quickly bring you up to speed in writing C# programs.

Course Objectives:

- Acquire a working knowledge of C# programming
- Learn about important interactions between C# and the .NET Framework
- Learn how to implement simple GUI programs using Windows Forms
- Gain a working knowledge of dynamic data type, named and optional arguments, and other new features in C# 4.0.

Audience: Programmers who need to design and develop C# for the .NET framework.

Prerequisites: Programming in a high-level language is required and some background in object-oriented programming is recommended.

Number of Days: 2 days

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. .NET: What You Need to Know
 Getting Started
 .NET: What Is Really Happening
 .NET Programming in a Nutshell
 Viewing the Assembly
 Viewing Intermediate Language
 Understanding .NET
 Visual Studio 2010
 Creating a Console Application
 Adding a C# File
 Using the Visual Studio Text Editor
 IntelliSense
 Build and Run the Project
 Pausing the Output | <ol style="list-style-type: none"> 2. C# Overview for the Sophisticated Programmer
 Visual C# and GUI Programs
 .NET Documentation
 Hello, World
 Compiling, Running (Command Line)
 Program Structure
 Namespaces
 Variables
 Input in C#
 More about Classes
 InputWrapper Class
 Input Wrapper Implementation
 Compiling Multiple Files |
|---|--|

- Control Structures
- switch
- C# Operators
- Precedence Table
- Types in C#
- Simple Types
- Types in System Namespace
- Integer Data Types
- Floating Point Data Types
- Implicit Conversions
- Explicit Conversions
- Boolean Data Type
- struct
- Uninitialized Variables
- Enumeration Types
- Nullable Types
- Reference Types
- Class Types
- object
- string
- Copying Strings
- StringBuilder Class
- Classes and Structs
- Static and Instance Methods
- Method Parameters
- No “Freestanding” Functions in C
- Classes with All Static Methods
- Parameter Passing
- Parameter Terminology
- Value Parameters
- Reference Parameters
- Output Parameters
- Structure Parameters
- Class Parameters
- Method Overloading
- Variable Length Parameter Lists
- Arrays
- One-Dimensional Arrays
- System.Array
- Jagged Arrays
- Rectangular Arrays
- foreach for Arrays
- Boxing and Unboxing
- Implicitly Typed Variables
- Output in C#
- Formatting

- Exceptions
- Checked Integer Arithmetic
- Throwing New Exceptions
- finally
- System.Exception
- 3. Object-Oriented Programming in C#**
- C# Object-Oriented Features
- Encapsulation and Accessors
- Asymmetric Accessor Accessibility
- Using a Property
- Indexers
- Visual Studio Console Projects
- Files in Skeleton Solution
- Source Files
- Account Class
- Constructors
- Static Members
- Static in Main
- Static Constructor
- Constant and Readonly Fields
- Auto-Implemented Properties
- Inheritance in C#
- New Version of Base Class
- Features of the New Base Class
- Derived Class
- Overriding a Virtual Function
- Abstract Classes
- Keyword: abstract
- Derived Class
- Sealed Classes
- Access Control and Assemblies
- Internal Accessibility
- Class Libraries
- 4. C# and the .NET Framework**
- Components and OO in C#
- Interfaces in C#
- Interface Inheritance
- Programming with Interfaces
- Implementing Interfaces
- Using an Interface
- Dynamic Use of Interfaces
- is Operator
- as Operator
- Resolving Ambiguity
- Explicit Interface Implementation
- System.Object

- Collections
- ArrayList
- ArrayList Methods
- IEnumerable and IEnumerator
- Using Enumerators
- Collections of User-Defined Objects
- Account Class
- AccountList Class
- Copy Semantics in C#
- Deep Copy and ICloneable
- Writing Generic Code
- Using a Class of object
- Generic Types
- Generic Syntax in C#
- Generic Client Code
- System.Collections.Generic
- Object Initializers
- Collection Initializers
- Anonymous Types
- Attributes
- 5. Delegates and Events**
- Callbacks and Delegates
- Usage of Delegates
- Declaring a Delegate
- Defining a Method
- Creating a Delegate Object
- Calling a Delegate
- Random Number Generation
- A Random Array
- Anonymous Methods
- Combining Delegate Objects
- Account.cs
- DelegateAccount.cs
- Lambda Expressions
- Named Method
- Anonymous Method
- Events
- Events in C# and .NET
- Client Side Event Code
- 6. Introduction to Windows Forms**
- Creating a Windows Forms App
- Partial Classes
- Windows Forms Event Handling
- Add Events for a Control
- Events Documentation
- Closing a Form
- ListBox Control
- 7. New Features in C# 4.0**
- dynamic Type
- dynamic versus object
- Behavior of object
- Behavior of dynamic
- Named Arguments
- Optional Arguments
- Book Class
- Using Optional Arguments
- Automating Office with C# 4.0
- Automating Excel
- Automating Word
- Variance in Generic Interfaces
- Variance with IComparer<T>
- Interfaces with Variance Support
- 8. Appendix A – Using Visual Studio 2010**
- A Visual Studio Solution
- Toolbars
- Customizing a Toolbar
- Creating a Console Application
- Adding a C# File
- Using the Visual Studio Text Editor
- Build and Run the Bytes Project
- Running the Bytes Project
- Executable File Location
- Managing Configurations
- Project Configurations
- Creating a New Configuration
- Setting Configuration Build Settings
- Debugging
- Breakpoints
- Watch Variables
- Debug Toolbar
- Stepping with the Debugger
- Call Stack and Call Hierarchy
- Adding a Reference
- Project Dependencies
- Startup Project
- Hidden Files
- 9. Appendix B – Language Integrated Query (LINQ)**
- Language-Integrated Query (LINQ)
- Using IEnumerable<T>
- Basic LINQ Query Operators

Obtaining a Data Source

Filtering

Ordering

Aggregation

Obtaining Lists and Arrays

Deferred Execution

10. Appendix C – Learning Resources