

# Technical Training Solutions



## MICROSOFT C# .NET 2010 WITH VISUAL STUDIO 2010 FUNDAMENTALS (5 Day Course)

Technical  
Course  
Outline

### Course Summary

During this intensive course, students will be exposed to a wide range of topics that will lead to their proficiency with not only the C# .NET 2010 programming language, but also the Visual Studio 2010 development environment, and the Microsoft .NET 4.0 Framework.

While many different project/application types will be explored (Windows Forms, ASP .NET Web Application, ASP .NET Web Services, Class Libraries, Console), heavy emphasis is placed on class library development and object-oriented programming principles. Debugging techniques and the underlying workings of the .NET Framework are also covered in detail.

### Intended Audience

This course is intended for those who have **no prior .NET programming experience** and will use C# .NET with Visual Studio 2010 to develop Windows and web based applications.

### Prerequisites

- There are no required prerequisites for this course.

### Course Contents:

#### Getting To Know Visual Studio 2010

- The Various Types Of .NET Projects
- Project Configuration Settings ("My Project" Designer)
- Building, Saving, & Running Projects
- Customizing Visual Studio 2010 Settings
  - Compile Options
- Getting Help!
  - IntelliSense, Code Expansion, & Code Snippets
  - The MSDN Library & The "Object Browser" Window

#### Introducing The Microsoft .NET 4.0 Framework

- The CLR (Common Language Runtime)
  - The CLS (Common Language Specification)
  - The CTS (Common Type System)
- Assemblies, Namespaces & .NET Class Libraries
- Types
  - Value Types & Reference Types
  - Partial Types
- .NET Memory Management
  - The Stack & The Managed Heap
  - Garbage Collection
  - Casting vs. Converting vs. Parsing

#### Windows Forms Applications

- Windows Forms vs. ASP .NET (Web Forms) Applications
- Adding Controls & Setting Control Properties
- Incorporating Event Handling Code

#### Exception Handling, Debugging & Testing Code

- The **Exception** Class
- **try...catch...finally** Blocks
- Setting Breakpoints & Stepping Through Code
- The Various Debugging Windows

#### Assemblies

- Private vs. Shared Assemblies
- The End Of DLL Hell?
  - The Assembly Manifest, References, & Versioning
- The Global Assembly Cache (The GAC)
- Strong Naming An Assembly

#### C# Language Elements & Syntax

- Naming Conventions
- Using Variables & Primitive Types
  - Primitive Types & C# Variable Declarations
  - Verbatim Strings

#### C# Language Elements & Syntax (continued)

- Decision Making Structures
  - **if** & **switch** Constructs
- C# Operators
- Looping Structures
  - **for** & **while**
- C# Functions
  - Returning A Value vs. **void**
  - **ref** & **out** Parameters

#### Object-Oriented Programming & Building Class Libraries

- Core OO Concepts
  - Classes vs. Instances
  - Abstraction, Encapsulation, Inheritance, Polymorphism
  - Overloading, Virtual (Overriding), Static, Hiding
- Designing The Class
  - Fields, Properties, Methods, Events, Constructors
  - Determining Member Scope
  - Visual Studio 2010's "Class Diagram" & "Class Designer"
- Advanced Object-Oriented Programming Concepts
  - Abstract & Sealed Classes
  - Event Delegates
  - Building & Implementing Interfaces
  - Building Enumerable Types (Enums)

#### Utilizing Class Libraries

- The .NET Console Application Project Type
- References & Importing Namespaces
- Instancing Classes

#### Introduction To ASP .NET Web Development

- The HTTP Request/Response (Client/Server) Architecture Model
- ASP .NET Web **Sites** vs. ASP .NET Web **Applications**
- Web Form, HTML, & HTML Server Controls

#### Introduction To ASP .NET XML Web Services

- Building Web Services
  - The [**WebService**] and [**WebMethod**] Compiler Attributes
- Using The Visual Studio Web Service Test Bed
- The Web Service Description Language (WSDL) Document
- Building A Web Service Client
  - Making The Web Reference
  - Using The Web Proxy