

Visual C#.NET Introduction for Developers

5 day course

Overview

You have experience in programming in at least one other object-oriented or object-based programming language. You are now entering the new world of .NET programming, using the C# programming language and the Visual C#.NET integrated development environment inside Visual Studio.NET. This is one in a series of C# programming courses offered by Systematix.

Prerequisites

To ensure your success, we recommend you first take the following Systematix course or have equivalent knowledge Object-Oriented Programming Principles .NET Framework Introduction Visual Studio.NET Introduction

Benefits

.NET is a software revolution. As a programmer, you will want your skills on the leading edge of this revolution. Of all the languages targeting the .NET platform, C# has been tailor-made for future .NET developers.

Target student

Visual Basic, C++, and Java programmers with at least one year of practical experience who want to learn C# programming.

What's next

For further C# training then Visual C#.NET Advanced is next in the series. For developers of ASP applications ASP.NET Using Visual C#, ADO.NET, and XML is the next course in this series.

Course Content:

Creating Simple C# Programs

- Create a Simple C# Console Application
- Use a Namespace
- Comment Code
- Create a Simple Class Library
- Test a Simple Class Library
- Create a Simple C# Windows GUI Application

Working with Classes

- Define and Use a Class
- Use Access Modifiers
- Define Namespaces
- Declare an Abstract Class
- Derive a Concrete Class from an Abstract Class
- Use an Abstract Class and its Concrete Derived Class

Working with Methods

- Implement Constructors
- Implement a Destructor
- Override a Method
- Overload a Method
- Overload an Operator
- Pass Parameters

Programming with Forms and Controls

- Create a Graphical Program Using Standard Built-in Controls
- Create a Custom Control
- Use Custom Controls on a Form

Writing Statements that Control Program Flow

- Write Conditional Statements
- Write Loop Statements
- Write Jump Statements

Using Types in C#

- Use Pre-defined Data Types
- Differentiate Between Reference and Value Types
- Use Casting
- Define and Use a Struct Type
- Define and Use an Enum Type

Programming with Exceptions

- Catch Exceptions
- Define Custom Exceptions
- Throw Exceptions
- Utilize the finally Keyword

Working with Interfaces

- Define an Interface
- Implement Interfaces in Derived Classes
- Invoke Interface Methods

Working with Arrays and Collections

- Define a Rectangular or Higher-Dimensional Array
- Use a Rectangular Array
- Define a Jagged Array
- Use a Jagged Array
- Choose and/or Create a Collection Type
- Use A Collection Type

Working with Properties and Indexers

- Expose Properties
- Expose Indexers

Building and Deploying Assemblies

- Create an Assembly with the C# Command Line Compiler
- Use Ildasm
- Deploy an Assembly Locally
- Create and Deploy a Shared Assembly

Interoperating with Legacy Code

- Write Code that Interoperates with Win32
 - Expose Legacy COM Components to C# Clients
 - Expose C# Components to Legacy COM Clients
-