

Visual C# .NET: Introduction for Developers

Course length: 5 days

Course Description

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.

Prerequisites: To ensure your success, we recommend you first take the following courses or have equivalent knowledge:

- Object-Oriented Programming: Principles
- .NET Framework: Introduction
- Visual Studio.NET: Introduction

Delivery Method: Instructor led, group-paced, classroom-delivery learning model with structured hands-on activities.

Benefits: The software development industry has gone through several revolutions over the last 50 years, the most recent of which is the introduction of Microsoft's .NET platform. The software industry has made major transitions in the past, including machine languages (0s and 1s), symbolic languages (COBOL and FORTRAN), procedural language (Pascal and C), and object-oriented languages (Smalltalk, C++, and Java), each with its own improvements in productivity and programming power. .NET is a similar 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.

Performance-Based Objectives

Lesson objectives help students become comfortable with the course, and also provide a means to evaluate learning. Upon successful completion of this course, students will be able to:

- Build and run simple C#-based programs.
- Define and use classes in C#.
- Define and use methods in C#.
- Program GUI Windows applications with forms and controls.
- Write statements that control program flow.
- Identify and use reference and value types in C#.
- Program with built-in and user-defined exceptions.
- Identify and use reference and value types in C#.
- Define and use arrays and collections.
- Expose and access properties and indexers.
- Build and deploy local and shared assemblies.
- Write code that interoperates with legacy Windows code.

Course Content

Lesson 1: Creating Simple C# Programs

Topic 1A: Create a Simple C# Console Application

Topic 1B: Use a Namespace



Topic 1C: Comment Code
Topic 1D: Create a Simple Class Library
Topic 1E: Test a Simple Class Library
Topic 1F: Create a Simple C# Windows GUI Application

Lesson 2: Working with Classes

Topic 2A: Define and Use a Class
Topic 2B: Use Access Modifiers
Topic 2C: Define Namespaces
Topic 2D: Declare an Abstract Class
Topic 2E: Derive a Concrete Class from an Abstract Class
Topic 2F: Use an Abstract Class and its Concrete Derived Class

Lesson 3: Working with Methods

Topic 3A: Implement Constructors
Topic 3B: Implement a Destructor
Topic 3C: Override a Method
Topic 3D: Overload a Method
Topic 3E: Overload an Operator
Topic 3F: Pass Parameters

Lesson 4: Programming with Forms and Controls

Topic 4A: Create a Graphical Program Using Standard Built-in Controls
Topic 4B: Create a Custom Control
Topic 4C: Use Custom Controls on a Form

Lesson 5: Writing Statements that Control Program Flow

Topic 5A: Write Conditional Statements
Topic 5B: Write Loop Statements
Topic 5C: Write Jump Statements

Lesson 6: Using Types in C#

Topic 6A: Use Pre-defined Data Types
Topic 6B: Differentiate Between Reference and Value Types
Topic 6C: Use Casting
Topic 6D: Define and Use a Struct Type
Topic 6E: Define and Use an Enum Type

Lesson 7: Programming with Exceptions

Topic 7A: Catch Exceptions
Topic 7B: Define Custom Exceptions
Topic 7C: Throw Exceptions
Topic 7D: Utilize the finally Keyword

Lesson 8: Working with Interfaces

Topic 8A: Define an Interface
Topic 8B: Implement Interfaces in Derived Classes
Topic 8C: Invoke Interface Methods

Lesson 9: Working with Arrays and Collections

Topic 9A: Define a Rectangular or Higher-Dimensional Array
Topic 9B: Use a Rectangular Array



Topic 9C: Define a Jagged Array
Topic 9D: Use a Jagged Array
Topic 9E: Choose and/or Create a Collection Type
Topic 9F: Use A Collection Type

Lesson 10: Working with Properties and Indexers

Topic 10A: Expose Properties
Topic 10B: Expose Indexers

Lesson 11: Building and Deploying Assemblies

Topic 11A: Create an Assembly with the C# Command Line Compiler
Topic 11B: Use Ildasm
Topic 11C: Deploy an Assembly Locally
Topic 11D: Create and Deploy a Shared Assembly

Lesson 12: Interoperating with Legacy Code

Topic 12A: Write Code that Interoperates with Win32
Topic 12B: Expose Legacy COM Components to C# Clients
Topic 12C: Expose C# Components to Legacy COM Clients

