

Course Outline | C# for Existing Developers

4 day(s)

Microsoft C Sharp for Existing Developers is a definite must for any person using another language like VB or C++. Our tutor has knowledge of a number of languages and often adds pointers for VB/C++ when there is something to note.

Prerequisites

Previous knowledge and experience of developing applications with a programming language.

Course Outline

Welcome to C#

- The Visual Studio Environment
- Writing Your First Program
- Using Namespaces
- Creating a Graphical Application

Working with Variables, Operators, and Expressions

- Understanding Statements
- Using Identifiers
- Using Variables
- Working with Primitive Data Types
- Incrementing and Decrementing Variables
- Declaring Implicitly Typed Local Variables

Writing Methods and Applying Scope

- Creating Methods
- Applying Scope
- Writing Methods
- Using Optional Parameters and Named Arguments

Using Decision Statements

- Declaring Boolean Variables
- Using Boolean Operators
- Using if Statements to Make Decisions
- Using switch Statements

Using Compound Assignment and Iteration Statements

- Using Compound Assignment Operators
- Writing while Statements
- Writing for Statements
- Writing do Statements

Managing Errors and Exceptions

- Coping with Errors
- Trying Code and Catching Exceptions
- Using Checked and Unchecked Integer Arithmetic
- Throwing Exceptions
- Using a finally Block

Understanding the C# Language

- Creating and Managing Classes and Objects
- Understanding Classification

Course Outline | C# for Existing Developers

- The Purpose of Encapsulation
- Defining and Using a Class
- Controlling Accessibility
- Understanding static Methods and Data

Understanding Values and References

- Copying Value Type Variables and Classes
- Understanding Null Values and Nullable Types
- Using ref and out Parameters
- How Computer Memory Is Organized
- The System.Object Class
- Boxing
- Unboxing
- Casting Data Safely

Creating Value Types with Enumerations and Structures

- Working with Enumerations
- Working with Structures

Using Arrays and Collections

- What Is an Array?
- What Are Collection Classes?
- Understanding Parameter Arrays
- Using Array Arguments
- Comparing Parameters Arrays and
- Optional Parameters

Working with Inheritance

- What Is Inheritance?
- Using Inheritance
- Understanding Extension Methods

Creating Interfaces and Defining

- Abstract Classes
- Understanding Interfaces
- Abstract Classes
- Sealed Classes

Using Garbage Collection and Resource Management

- The Life and Times of an Object
- Resource Management
- Implementing Exception-Safe Disposal

Creating Components

- Implementing Properties to Access Fields
- Implementing Encapsulation by Using Methods
- What Are Properties?
- Understanding the Property Restrictions
- Declaring Interface Properties
- Generating Automatic Properties
- Initializing Objects by Using Properties

Interrupting Program Flow and Handling Events

- Declaring and Using Delegates

Course Outline | C# for Existing Developers

- Lambda Expressions and Delegates
- Enabling Notifications with Events

Introducing Generics

- The Problem with objects
- The Generics Solution
- Creating a Generic Class
- Creating a Generic Method
- Variance and Generic Interfaces

Enumerating Collections

- Enumerating the Elements in a Collection
- Implementing an Enumerator by Using an Iterator

Querying In-Memory Data by Using Query Expressions

- What Is Language Integrated Query?
- Using LINQ in a C# Application

Querying Information in a Database

- Querying a Database by Using ADO.NET
- Querying a Database by Using LINQ to SQL

Displaying and Editing Data by Using the Entity Framework and Data Binding

- Using Data Binding with the Entity Framework
- Using Data Binding to Modify Data