



## Course 921 – C# Programming

Duration: 5 days

### You Will Learn How To

- Write and compile C# applications using Visual Studio
- Understand different C# constructs and other language features
- Understand Object-Oriented Programming concepts
- Reuse existing classes using inheritance
- Create simple GUI applications using multi-layered architecture
- Take advantage of the built-in libraries of the .NET framework
- Access databases from C# applications

### Course Benefits

C# is the primary language used to create Windows applications. This C# Programming course will teach you object-oriented programming concepts and use C# to build applications on top of the Microsoft .NET framework. In this course, you will learn how to take advantage of C# to create a wide range of Windows applications.

### Who Should Attend

This C# programming course assumes prior programming experience using any language such as C or Visual Basic. Students are expected to be familiar with concepts like variables, data types, loops, and arrays. This course covers OOP extensively so knowing OOP is optional. Knowing HTML or SQL alone is not sufficient.

### Course Content

#### Introduction to C# Programming

- History
- What is C#?
- Introduction to the .NET Framework
- Common Language Runtime
- Common Intermediate Language
- Managed Assemblies
- Base Class Libraries
- C# Development Process

#### C# Development Overview

- Your First C# Program
- Using Visual Studio
- Creating a New C# Project
- Projects and Solutions
- Writing Your Code
- Running the Application

#### C# Language Fundamentals

- Elements of a C# Program
- Identifiers
- Literals and Data Types
- Operators

#### Getting Input from the Keyboard

- Input via Console Application
- Input via GUI
- Converting Strings to Integers

### Control Flow Structures

- Control Structures
- Decision Control Structures – if, if-else, and nested if and if-else
- switch Statement
- Repetition Control Structures – while, do-while, and for
- Branching Statements – break and continue

### Arrays

- Introduction to Arrays
- Declaration of Arrays
- Array Construction and Instantiation
- Accessing Array Elements
- Array Initialization
- Iterating Through an Array
- Arrays as Objects
- Jagged Arrays
- Multi-Dimensional Arrays
- Command-line Arguments

### Object-Oriented Programming

- What is an Object?
- Encapsulation
- Classes
- Object References
- Declaring Reference Types
- Creating Objects
- Constructors
- Classes vs Objects
- Accessing Properties
- Calling Methods and instance Methods
- Static Methods
- Pass-by-Value Parameter Passing

- Pass-by-Reference Parameter Passing
- Comparing Objects
- String Interning

### Creating Your Own Classes

- Declaring Classes
- .NET Class Attributes
- Declaring Fields
- Declaring Properties
- Class Variables or Static Attributes
- Static Fields and Properties
- Declaring Methods
- Returning Values
- Static Methods
- Variable Scope
- Attributes vs Local Variables
- “this” Reference and Usage
- Method Overloading
- Constructors
- Object Initializers
- Destructors
- “enum” Types
- “struct” Types
- Namespaces
- Adding References

### Inheritance, Polymorphism, and Interfaces

- What is Inheritance?
- Property and Method Overriding
- The base Keyword
- Sealed Classes and Methods
- Constants
- Abstract Classes
- Interfaces



- Polymorphism
- Object Conversion and Casting
- Access Modifiers
- Extension Methods

### Handling Exceptions

- Exceptions
- Catching Exceptions
- “try-catch-finally” Block
- The Call Stack
- How Exceptions are Caught
- Forwarding an Exception through the Call Stack
- User-Defined Exceptions
- Throwing Exceptions: The “throw” Keyword

### Essential C# Classes

- The “Object” Class
- The “Math” Class
- How to Generate Random Numbers
- The “String” and the “StringBuilder” Class
- The “DateTime” and “TimeSpan” Structures
- How to Format Numbers and Dates
- Collection Classes
- Generics
- Lambda Expressions

### Creating Desktop Applications Using WPF

- WPF and XAML
- Windows and Panels
- Elements and Controls
- Data Binding

- Events

### Accessing Databases Using ADO .NET

- Data Providers
- Main ADO .NET Classes
- Basic Steps in Using ADO .NET
- Navigating through the Data
- Reading Data Values from “SqlDataReader”
- Examples
- Creating, Updating, and Deleting Data in ADO .NET
- Inserting, Updating, and Deleting Data
- Using Parameterized SQL Statements
- SQL Injection
- Prepared and Parameterized Statements
- Mapping of SQL Types fo C# Types

### About ActiveLearning, Inc.

ActiveLearning is the Philippines' leading provider of Information Technology and Project Management education, where thousands of students take courses from Application Development to Project Management to Network Security, and much more. Our courses are taught by expert instructors, and learning is enhanced through a blend of in-depth lectures, workshops, and hands-on exercises.