www.logicalimagination.com
800.657.1494

Object-Oriented Programming in C#

Course #: VC-110 **Duration:** 5 days

Prerequisites

The student should have programming experience in a high-level language.

Details

This thorough and comprehensive course is a practical introduction to programming in C#, utilizing the services provided by .NET. It is current to Visual Studio 2022, .NET 6, and C# 10. Important newer features such as dynamic data types, named and optional arguments, tuples, asynchronous programming keywords, nullable reference types, record types, and top-level statements are covered. Supplements provide a tutorial on Visual Studio 2022, an overview of LINQ, and coverage of unsafe code and pointers in C#.

This course is intended to be fully accessible to programmers who do not already have a strong background in object-oriented programming in C-like languages, such as C++ or Java. It is ideal, for example, for procedural programmers who desire to learn C#.

An essential thrust of the course is to teach C# programming from an object-oriented perspective. It is often difficult for programmers trained originally in a procedural language to start "thinking in objects." This course introduces object-oriented concepts early, and C# is developed in a way that leverages its object orientation. A case study illustrates creating a complete system using C# and .NET. Besides supporting traditional object-oriented features, such as classes, inheritance, and polymorphism, C# introduces several additional features, such as properties, indexers, delegates, events, and interfaces that make C# a perfect language for developing object-oriented and component-based systems.

C# as a language is elegant and powerful. But to utilize its capabilities fully, you need to have a good understanding of how it works with the .NET Framework. The course explores several significant interactions between C# and the .NET Framework, and it includes an introduction to major classes for collections, delegates, and events. It includes a succinct introduction to creating GUI programs using Windows Forms.

Software Needed

Microsoft Visual Studio 2022 and Windows 10 or higher. The free Visual Studio Community 2022 can be used.

Outline

Object-Oriented Programming in C#

- **Introduction to .NET**
  - What is .NET?
  - .NET Framework, .NET Core, and .NET 6
  - Application Models
  - Managed Code
  - Visual Studio 2022
  - Console Programs and New Console Template
  - GUI Programs

- **First C# Programs**
  - Hello, World
  - Namespaces
  - Variables and Expressions
  - Using C# as a Calculator
  - Input/Output in C#
  - .NET Class Library
- **Data Types in C#**
  - Data Types
  - Integer Types
  - Floating Point Types
  - Decimal Type
  - Characters and Strings
  - Boolean Type
  - Conversions
  - Nullable Types
- **Operators and Expressions**
  - Operator Cardinality
  - Arithmetic Operators
  - Relational Operators
  - Logical Operators
  - Bitwise Operators
  - Assignment Operators
  - Expressions
  - Checked and Unchecked
- **Control Structures**
  - If Tests
  - Loops
  - Arrays
  - Foreach
  - More about Control Flow
  - Switch
- **Object-Oriented Programming**
  - Objects
  - Classes
  - Inheritance
  - Polymorphism
  - Object-Oriented Languages
  - Components
- **Classes**
  - Classes as Structured Data
  - Methods
  - Constructors and Initialization
  - Static Fields and Methods
  - Constant and Readonly
- **More about Types**
  - Overview of Types in C#
  - Value Types
  - Boxing and Unboxing
  - Reference Types
  - Implicitly Types Variables
- **Methods, Properties, and Operators**
  - Methods
  - Parameter Passing
  - Method Overloading
  - Variable-Length Parameter Lists
  - Properties
  - Auto-Implemented Properties
  - Operator Overloading
- **Characters and Strings**
  - Characters
  - Strings
  - String Input
  - String Methods
  - StringBuilder Class

    - Programming with Strings
- **Arrays and Indexers**
    - Arrays
    - System.Array
    - Random Number Generation
    - Jagged Arrays
    - Rectangular Arrays
    - Arrays as Collections
    - Indexers
- **Inheritance**
    - Single Inheritance
    - Access Control
    - Method Hiding
    - Initialization
- **Virtual Methods and Polymorphism**
    - Virtual Methods and Dynamic Binding
    - Method Overriding
    - Fragile Base Class Problem
    - Polymorphism
    - Abstract Classes
    - Sealed Classes
    - Heterogeneous Collections
- **Formatting and Conversion**
    - ToString
    - Format Strings
    - String Formatting Methods
    - Type Conversions
- **Exceptions**
    - Exception Fundamentals
    - Structured Exception Handling
    - User-Defined Exception Classes
    - Inner Exceptions
- **Interfaces**
    - Interface Fundamentals
    - Programming with Interfaces
    - Using Interfaces at Runtime
    - Resolving Ambiguities
- **.NET Interfaces and Collections**
    - Collections
    - IEnumerable and IEnumerator
    - Copy Semantics and ICloneable
    - Comparing Objects
    - Generic Types
    - Type-Safe Collections
    - Object Initializers
    - Collection Initializers
    - Anonymous Types
- **Delegates and Events**
    - Delegates
    - Anonymous Methods
    - Lambda Expressions
    - Events
- **Introduction to Windows Forms**
    - Creating Windows Applications
    - Partial Classes
    - Buttons, Labels, and Textboxes
    - Handling Events
    - Listbox Controls
- **Newer Features in C#**
    - Dynamic Data Type
    - Named and Optional Arguments
    - Variance in Generic Interfaces
    - Asynchronous Programming Keywords
    - Nullable Reference Types
    - Record Types

- Top-Level Statements

Supplemental Topics

- **Using Visual Studio 2022**
  - Signing in to Visual Studio
  - Overview of Visual Studio 2022
  - Creating a Console Application
  - Project Configurations
  - Debugging
  - Multiple-Project Solutions
- **Language Integrated Query (LINQ)**
  - What Is LINQ?
  - Basic Query Operators
  - Filtering
  - Ordering
  - Aggregation
- **Unsafe Code and Pointers in C#**
  - Unsafe Code
  - C# Pointer Type