Object-Oriented Programming in C#, Rev. 6.1

Course Summary

Description

Microsoft .NET is an advance in programming technology that greatly simplifies application development, both for traditional, proprietary applications and for the emerging paradigm of Web-based services. .NET Framework is the original implementation of .NET, running on Windows. .NET Core is a new package-based implementation that is cross-platform, running on Mac and Linux besides Windows. .NET 6.1 is the next version of .NET Core and is the main implementation of .NET going forward. .NET Framework 4.x continues to be supported.

Part of this technology is the new language from Microsoft, C#. This language combines the power of C++ and the ease of development of Visual Basic 6. It bears a striking resemblance to Java and improves on that language. C# has become the dominant language for building new applications on Microsoft platforms.

This thorough and comprehensive course is a practical introduction to programming in C#, utilizing the services provided by .NET. This course emphasizes the C# language. It is current to Visual Studio 2019, .NET 6.1 and C# 9.0. Important newer features such as dynamic data types, named and optional arguments, tuples, asynchronous programming keywords, nullable reference types, and immutable record types are covered. Supplements provide a tutorial on Visual Studio 2019, 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 important 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 is used to illustrate 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 compelling language for developing object-oriented and component-based systems. This course provides thorough coverage of all these features.

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 important 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. The course concludes with a chapter covering the newer features in the language through C# 9.0.

Numerous programming examples and exercises are provided, including the case study. The student will receive a comprehensive set of materials, including course notes and all the programming examples.

Object-Oriented Programming in C#, Rev. 6.1

Course Summary (cont'd)

Objectives

At the end of this course, students will be able to:

- Acquire a working knowledge of C# programming
- Learn how to implement programs using C# and classes from the .NET Framework
- Gain an understanding of the object-oriented programming paradigm
- Learn how to implement simple GUI programs using Windows Forms
- Gain a working knowledge of important newer features in C#

Topics

- Introduction to NET
- First C# Programs
- Data Types in C#
- Operators and Expressions
- Control Structures
- Object-Oriented Programming
- Classes
- More about Types
- Methods, Properties and Operators
- Characters and Strings
- Arrays and Indexers
- Inheritance
- Virtual Methods and Polymorphism
- Formatting and Conversion
- Exceptions
- Interfaces
- .NET Interfaces and Collections
- Delegates and Events
- Introduction to Windows Forms
- Newer Features in C#
- Appendix A. Learning Resources
- Supplement 1. Using Visual Studio 2019
- Supplement 2. Language Integrated Query (LINQ)
- Supplement 3. Unsafe Code and Pointers in C#

Audience

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#.

Prerequisites

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

Duration

Five days

Object-Oriented Programming in C#, Rev. 6.1

Course Outline

I. Introduction to NET
A. What is .NET?
B. .NET Framework, NET Core and .NET 6.1
C. Application Models
D. Managed Code
E. Visual Studio 2019
F. C# Console and GUI Programs

II. First C# Programs
A. Hello, World
B. Namespaces
C. Variables and Expressions
D. Using C# as a Calculator
E. Input/Output in C#
F. .NET Framework Class Library

III. Data Types in C#
A. Data Types
B. Integer Types
C. Floating Point Types
D. Decimal Type
E. Characters and Strings
F. Boolean Type
G. Conversions
H. Nullable Types

IV. Operators and Expressions
A. Operator Cardinality
B. Arithmetic Operators
C. Relational Operators
D. Logical Operators
E. Bitwise Operators
F. Assignment Operators
G. Expressions
H. Checked and Unchecked

V. Control Structures
A. If Tests
B. Loops
C. Arrays
D. Foreach
E. More about Control Flow
F. Switch

VI. Object-Oriented Programming
A. Objects
B. Classes
C. Inheritance
D. Polymorphism
E. Object-Oriented Languages
F. Components

VII. Classes
A. Classes as Structured Data
B. Methods
C. Constructors and Initialization
D. Static Fields and Methods
E. Constant and Readonly

VIII. More about Types
A. Overview of Types in C#
B. Value Types
C. Boxing and Unboxing
D. Reference Types
E. Implicitly Typed Variables

IX. Methods, Properties and Operators
A. Methods
B. Parameter Passing
C. Method Overloading
D. Variable-Length Parameter Lists
E. Properties
F. Auto-Implemented Properties
G. Operator Overloading

X. Characters and Strings
A. Characters
B. Strings
C. String Input
D. String Methods
E. StringBuilder Class
F. Programming with Strings

XI. Arrays and Indexers
A. Arrays
B. System.Array
C. Random Number Generation
D. Jagged Arrays
E. Rectangular Arrays
F. Arrays as Collections
G. Bank Case Study—Step 1
H. Indexers

Object-Oriented Programming in C#, Rev. 6.1

Course Outline (cont'd)

XII. Inheritance
A. Single Inheritance
B. Access Control
C. Method Hiding
D. Initialization
E. Bank Case Study—Step 2

XIII. Virtual Methods and Polymorphism
A. Virtual Methods and Dynamic Binding
B. Method Overriding
C. Fragile Base Class Problem
D. Polymorphism
E. Abstract Classes
F. Sealed Classes
G. Heterogeneous Collections
H. Bank Case Study—Step 3

XIV. Formatting and Conversion
A. ToString
B. Format Strings
C. String Formatting Methods
D. Bank Case Study—Step 4
E. Type Conversions

XV. Exceptions
A. Exception Fundamentals
B. Structured Exception Handling
C. User-Defined Exception Classes
D. Inner Exceptions
E. Bank Case Study—Step 5

XVI. Interfaces
A. Interface Fundamentals
B. Programming with Interfaces
C. Using Interfaces at Runtime
D. Bank Case Study—Step 6
E. Resolving Ambiguities

XVII. .NET Interfaces and Collections
A. Collections
B. Bank Case Study—Step 7
C. IEnumerable and IEnumerator
D. Copy Semantics and ICloneable
E. Comparing Objects
F. Generic Types
G. Type-Safe Collections
H. Object Initializers
I. Collection Initializers
J. Anonymous Types
K. Bank Case Study—Step 8

XVIII. Delegates and Events
A. Delegates
B. Anonymous Methods
C. Lambda Expressions
D. Events

XIX. Introduction to Windows Forms
A. Creating Windows Applications Using Visual Studio 2019
B. Partial Classes
C. Buttons, Labels and Textboxes
D. Handling Events
E. Listbox Controls

XX. Newer Features in C#
A. Dynamic Data Type
B. Named and Optional Arguments
C. Variance in Generic Interfaces
D. Asynchronous Programming Keywords
E. New Features in C# 6.1 and C# 7.0
F. Nullable Reference Types in C# 8.0
G. Immutable Record Types in C# 9.0

XXI. Appendix A. Learning Resources
A. Electronic File Supplements

XXII. Supplement 1. Using Visual Studio 2019
A. Signing in to Visual Studio
B. Overview of Visual Studio 2019
C. Creating a Console Application
D. Project Configurations
E. Debugging
F. Multiple-Project Solutions

XXIII. Supplement 2. Language Integrated Query (LINQ)
A. What Is LINQ?
B. Basic Query Operators
C. Filtering
D. Ordering
E. Aggregation

XXIV. Supplement 3. Unsafe Code and Pointers in C#
A. Unsafe Code
B. C# Pointer Type