

# Object-Oriented Programming in C#

**Course No.**

9410

**Description**

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.

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. The final chapter provides a succinct introduction to creating GUI programs using Windows Forms.

**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 Visual Basic or COBOL programmers who desire to learn C#. The course may also be taken by more experienced programmers who desire a thorough introduction to C# with many example programs. It is structured so that more experienced programmers can cleanly skip the material they already know.

**Prerequisites**

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

**Objectives**

- Gain a basic understanding of the philosophy and architecture of .NET
- Acquire a working knowledge of C# programming
- Learn how to implement programs using C# and classes from the .NET Framework
- Learn how to implement simple GUI programs using Windows Forms

**Format**

Lecture and hands-on workshops

**System Requirements**

Course exercises require Microsoft .NET and Microsoft Visual Studio .NET on Windows 2000.

A good minimal hardware profile for this course would have a Pentium 500-MHz or equivalent CPU, 256 MB of RAM, and at least 500 MB of free disk space for tools installation and courseware.

**Duration**

5 days

## Course Contents

### Part 1. Introduction to .NET and C#

#### 1. NET Framework

- .NET: What You Need To Know
- What Is Microsoft .NET?
- .NET Framework Overview
- Common Language Runtime

#### 2. First C# Programs

- Hello, World
- Namespaces
- Variables and Expressions
- Using C# as a Calculator
- Input/Output in C#
- .NET Framework Class Library

#### 3. Visual Studio .NET

- Overview of Visual Studio .NET
- Creating a Console Application
- Project Configurations
- Debugging

### Part 2. C# as a Language in the C Family

#### 4. Simple Data Types

- Data Types
- Integer Types
- Floating Point Types
- Decimal Type
- Character Type
- Boolean Type
- Conversions

#### 5. Operators and Expressions

- Operator Cardinality
- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Expressions
- Checked and Unchecked

#### 6. Control Structures

- If Tests
- Loops
- Preview of Arrays and Foreach
- More About Control Flow
- Switch

### Part 3. C# Programming Fundamentals

#### 7. Object-Oriented Programming

- Objects
- Classes
- Inheritance
- Polymorphism
- Object-Oriented Languages
- Components
- Electronic Commerce Game Case Study

#### 8. Classes

- Classes as Structured Data
- Methods
- Constructors and Initialization
- Static Fields and Methods
- Constant and Read-only

#### 9. The C# Type System

- Overview of Types in C#
- Value Types
- Boxing and Unboxing
- Reference Types

#### 10. Methods, Properties, and Operators

- Methods
- Parameter Passing
- Method Overloading
- Variable-Length Parameter Lists
- Properties
- Operator Overloading

#### 11. Characters and Strings

- Characters
- Strings
- String Input
- String Methods
- StringBuilder Class
- Programming with Strings

## 12. Arrays and Indexers

- Arrays
- System.Array
- Random Number Generation
- Jagged Arrays
- Rectangular Arrays
- Arrays as Collections
- Bank Case Study—Step 1
- Indexers

## 13. Inheritance

- Single Inheritance
- Access Control
- Method Hiding
- Initialization
- Bank Case Study—Step 2

## 14. Virtual Methods and Polymorphism

- Virtual Methods and Dynamic Binding
- Method Overriding
- Fragile Base Class Problem
- Polymorphism
- Abstract Classes
- Sealed Classes
- Heterogeneous Collections
- Bank Case Study—Step 3

## 15. Formatting and Conversion

- ToString
- Format Strings
- String Formatting Methods
- Bank Case Study—Step 4
- Type Conversions

## 16. Exceptions

- Exception Fundamentals
- Structured Exception Handling
- User-Defined Exception Classes
- Inner Exceptions
- Bank Case Study—Step 5

## 17. Interfaces

- Interface Fundamentals
- Programming with Interfaces
- Using Interfaces at Runtime
- Bank Case Study—Step 6
- Resolving Ambiguities

## Part 4. C# and the .NET Framework

### 18. Interfaces and the .NET Framework

- Collections
- Bank Case Study—Step 7
- Copy Semantics and ICloneable
- Comparing Objects
- Understanding Frameworks

### 19. Delegates and Events

- Delegates
- Stock Market Simulation
- Events

### 20. Introduction to Windows Forms

- Creating Windows Applications Using Visual Studio .NET
- Handling Events

---

#### About Keane

Keane partners with businesses and government agencies to *optimize* IT investments by delivering exceptional evolution, operation, and maintenance of mission-critical systems and business processes. A US company with a large offshore capability, Keane combines local knowledge and local senior leadership with scalable global delivery that results in low-risk, actionable, cost-effective services and solutions – and a partnership that feels like an extension of your organization.

In business since 1965, Keane is an agile, full-service IT services firm headquartered in the United States with approximately 12,000 employees globally. For more information on Keane's services, solutions, products, and locations, please visit [www.keane.com](http://www.keane.com).