

Object-Oriented Programming in C# Course Summary

Description

Microsoft's .NET is a revolutionary advance in programming technology that greatly simplifies application development and is a good match for the emerging paradigm of Web-based services, as opposed to proprietary applications. Part of this technology is a new language, C#. This new language combines the power of C++ and the ease of development of Visual Basic. It bears a striking resemblance to Java and improves on that language. C# may well become the dominant language for building 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 2008 and .NET 3.5, which introduces important new features such as implicit typing, extension methods, lambda expressions and Language-Integrated Query (LINQ). The new features are covered in a new chapter.

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 new features in C# 3.0.

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
- Learn how to implement simple GUI programs using Windows Forms
- Gain a working knowledge of implicit typing, object initializers, anonymous types, extension methods, lambda expression, LINQ, and other new features in C# 3.0.

Topics

- NET: What You Need To Know
- 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
- New Features in C# 3.0

Audience

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 2008 and .NET 3.5, which introduces important new features such as implicit typing, extension methods, lambda expressions and Language-Integrated Query (LINQ). The new features are covered in a new chapter.

Prerequisites

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

Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Object-Oriented Programming in C#

Course Outline

- I. NET: What You Need To Know**
 - A. .NET Executables and the CLR
 - B. A .NET Testbed for C# Programming
 - C. Using Visual Studio 2005
- 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
- IX. Methods, Properties and Operators**
 - A. Methods
 - B. Parameter Passing
 - C. Method Overloading
 - D. Variable-Length Parameter Lists
 - E. Properties
 - F. 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#

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. Bank Case Study—Step 8

XVIII. Delegates and Events

- A. Delegates
- B. Anonymous Methods
- C. Events

XIX. Introduction to Windows Forms

- A. Creating Windows Applications Using Visual Studio 2005
- B. Partial Classes
- C. Buttons, Labels and Textboxes
- D. Handling Events
- E. Listbox Controls

XX. New Features in C# 3.0

- A. Auto-Implemented Properties
- B. Implicitly Typed Variables
- C. Object Initializers
- D. Collection Initializers
- E. Anonymous Types
- F. Partial Methods
- G. Extension Methods
- H. Lambda Expressions
- I. Language-Integrated Query (LINQ)