

Object-Oriented Programming in C# (VS2005) (5 Days) CTI 410

This course teaches C# programming from an object-oriented perspective, for programmers who do not already have a strong background in object-oriented languages, or programmers who desire a more thorough introduction to C#.

Prerequisites: Programming experience in a high-level language.

Minimum software requirements: Microsoft Windows XP Service Pack 2, Visual Studio 2005. Visual C# 2005 Express Edition can also be used.

Minimum hardware requirements for all machines: Pentium PC running at 1 GHz; 512 Mb RAM; 4GB free hard disk space before installation; Internet access; machines must be networked.

Microsoft PowerPoint and Internet access on instructor's workstation for presentation purposes.

Module 1: .NET: What You Need to Know

- .NET Executables and the CLR
- A .NET Testbed for C# Programming
- Using Visual Studio .NET

Module 2: First C# Programs

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

Module 3: Data Types in C#

- Data Types
- Integer Type
- Floating Point Type
- Decimal Type
- Characters and Strings
- Boolean Type
- Conversions
- Nullable Types

Module 4: Operators and Expressions

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

Module 5: Control Structures

- If Tests
- Loops
- Arrays
- Foreach
- More about Control Flow
- Switch

Module 6: Object-Oriented Programming

- Objects
- Classes
- Inheritance
- Polymorphism
- Object-Oriented Language Components

Module 7: Classes

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

Module 8: More About Types

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

Module 9: Methods, Properties and Operators

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

Module 10: Characters and Strings

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

Module 11: Arrays and Indexers

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

Module 12: Inheritance

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

Module 13: 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

Module 14: Formatting and Conversion

- ToString
- Format Strings
- String Formatting Methods
- Bank Case Study – Step 4
- Type Conversion

Object-Oriented Programming in C# (VS2005) (CTI 410) - continued

Module 15: Exceptions

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

Module 16: Interfaces

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

Module 17: .NET Interfaces and Collections

- Collections
- Bank Case Study – Step 7
- IEnumerable and IEnumerator
- Copy Semantics and ICloneable
- Comparing Objects
- Generic Types
- Type-Safe Collections
- Bank Case Study – Step 8

Module 18: Delegates and Events

- Delegates
- Anonymous Methods
- Events

Module 19: Introduction to Windows Forms

- Creating Windows Applications
Using Visual Studio 2005
- Partial Classes
- Buttons, Labels and Textboxes
- Handling Events
- Listbox Controls

Appendix A. Learning Resources

Electronic File supplements:

Supplement 1: Using Visual Studio 2005

- Overview of Visual Studio 2005
- Creating a Console Application
- Project Configurations
- Debugging
- Multiple-Project Solutions

Supplement 2: Unsafe Code and Pointers in C#

- Unsafe Code
- C# Pointer Type