

Course Description: This thorough and comprehensive course is a practical introduction to programming in C# utilizing the services provided by .NET. 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.

Who Should Attend: Those interested in learning about Microsoft's latest endeavor in the development world should take notice of this class. It 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.

Prerequisites: Some programming background is required. Recommended introductory level experience in any of the following languages is also required: C, C++, Visual Basic or Java. Understanding object oriented programming concepts is helpful, but not required.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Create and compile applications using the free C# compiler.
- Understand the concepts of the language C#.
- Use C#'s built-in functionality to read/write byte streams.
- Use advanced C#-only features such as properties, indexers, and events.
- Create valid object oriented components in order to easily update code; and,
- Create independent classes with their own operators, methods, and attributes defined.

Course Outline:

Introduction

Structure of a C# program
Compiling and Executing
Commenting
Value-Type Data
Arrays
Reference-Type Data

Arrays

Overview
Creating Arrays
Command Line Argument
Multi-Dimensional Arrays

Control Statements

Introduction
Statement Blocks
Methods
if Statements
switch Statement
loop Statements
break and continue
return Statements

More Data Types

Structs
Interfaces
Delegates
Enumerations
Namespaces
Applications and Libraries

C# Concepts

Application Startup
Member Access
Overloading
Scope
Value vs. Reference Parameters
Conversions
Exceptions

Classes

Object Oriented Programming
Simple Class Example
Class Methods
Constructors
Access Levels
Other Modifiers
Fields
Properties

Operators
Destructors
Static Constructor

Operators

The new operator
The typeof operator
Checked and Unchecked
Unary Operators
Binary Operators
Overloading the Multiplication Operator
Conditional Operator
Casting
Conversions

Inheritance

Introduction
Base Class Modifiers
Direct Inheritance
Static vs. non-static
Abstract Classes
Interfaces
Base Class Interfaces
Multiple Inheritance

Delegates And Events

Overview
Delegates
Events

Data Streams

Introduction
Streams
Readers
Writers
File I/O

Properties, Indexers And Attributes

Where to use
Properties
Indexers
Attributes

Threads

Overview
Understanding Threads
Using Threads in C#
Thread Properties
Managing Threads
Interrupting and Terminating Threads

Synchronization