

Learning to Program Using Visual C# 2010

Course Summary

Description

In this course, you'll learn to use Visual Studio 2010 to explore the Visual C# language. The course starts with a quick overview of the .NET platform, examining assemblies, Microsoft Intermediate Language, Visual Studio profiles, XML comments, IntelliSense, and debugging. From there, you'll learn all the language features that you must internalize in order to create full-featured Web or Windows applications that make best use of the .NET platform. You'll learn about data types, variables, and operators, along with all the important flow control structures. You'll work through several examples demonstrating the power of the .NET Framework, and dig into creating and consuming your own classes and objects. The course moves on to working with data structures, such as arrays and collection classes, before finishing up with discussions of generics, handling exceptions and working with delegates and events. The course concludes by introducing the new LINQ-oriented features including anonymous types, lambda expressions, and more. By the end of this course, you will understand the important basic concepts that will allow you to start creating the applications you need.

Objectives

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

- Create Console Application projects in Visual Studio .NET 2010.
- Convert a variable from one data type to another.
- Work with dates and times.
- Repeat blocks of code using looping statements.
- Generate and test your own classes using the Class Designer and Object Test Bench tools.
- Use methods in your classes.
- Control how derived classes inherit from base classes.
- Create classes containing arrays, adding support for indexers and enumerators.
- Use delegate types to provide flexibility and type safety.
- Create generic classes and methods.
- Use Exception objects to determine what error has occurred.
- Create your own custom collection classes.
- Use anonymous types, lambda expressions, extension methods, object initializers, and implicit type declarations.

Topics

- Getting Started
- Data Types and Variables
- Using the .NET Framework
- Branching and Flow Control
- Classes and Objects
- Properties and Methods
- Object-Oriented Techniques
- Working with Arrays
- Delegates and Events
- Generics
- Handling Exceptions
- Collection Classes
- LINQ

Learning to Program Using Visual C# 2010

Course Summary

Prerequisites

This course assumes that students have some programming background. No specific experience with Visual Studio 2010 or the .NET Framework is required. As with any such course, the more experience you bring to the course, the more you'll get out of it. This course moves quickly through a broad range of programming topics, but it does not require any prior .NET skills.

Duration

Three days

Learning to Program Using Visual C# 2010

Course Outline

I. Getting Started

- A. Learn the advantages and architecture of the .NET Framework
- B. Setup a development profile in Visual Studio .NET
- C. View the code that Visual Studio generates and understand what it does

II. Data Types and Variables

- A. Understand how to create variables and assign values
- B. Explore operators and see how they can be used to change values and compare expressions

III. Using the .NET Framework

- A. Using .NET Framework classes
- B. Explore basic file IO operations
- C. Learn how to work with strings

IV. Branching and Flow Control

- A. Making choices using conditional statements
- B. Manage flow control using branching statements
- C. Break out of loops when necessary

V. Classes and Objects

- A. Investigate .NET Framework classes to see how you can consume classes in your code
- B. Learn about properties, methods, and fields
- C. Create instances of classes using class constructors
- D. Investigate class details, including reference vs. value types, and more

VI. Properties and Methods

- A. Explore how to use properties in your classes
- B. Learn to control how property values are set and retrieved
- C. Understand how to pass arguments to methods
- D. Learn how to return both simple and complex data types from methods

VII. Object-Oriented Techniques

- A. Understand how derived classes inherit from base classes

- B. Explore how to add and modify members in derived classes
- C. Understand how to create and use interfaces
- D. Explore techniques for organizing your classes

VIII. Working with Arrays

- A. Create and initialize arrays
- B. Work with array methods and properties
- C. Investigate arrays in the .NET Framework
- D. Learn techniques for sorting arrays

IX. Delegates and Events

- A. Learn different ways to work with delegates
- B. Introduce anonymous delegates
- C. Learn how delegates and events are related
- D. Investigate events and event handlers

X. Generics

- A. Understand the advantages and benefits of generics
- B. Explore the use of generics to sort and search in arrays
- C. See how to use generic interfaces and constraints
- D. Explore the generic List class

XI. Handling Exceptions

- A. Learn to use try/catch blocks to handle runtime errors
- B. Throw exceptions back to procedure callers
- C. Use the finally block to run code unconditionally
- D. Create and handle user-defined exceptions

XII. Collection Classes

- A. Investigate the collection interfaces in the System.Collections.Generic namespace
- B. Create dynamically sized arrays using the List class
- C. Investigate the generic SortedDictionary, SortedList, Stack, and Queue classes

XIII. LINQ

- A. Motivate the need for LINQ
- B. Learn about the various LINQ providers
- C. Investigate simple LINQ to Objects, LINQ to SQL, and LINQ to XML samples