10266 Programming in C# with Microsoft Visual Studio 2010

Overview
The course focuses on C# program structure, language syntax, and implementation details with .NET Framework 4.0. This course describes the new enhancements in the C# 4.0 language by using Visual Studio 2010.

Prerequisite Comments
Before attending this course, students must have:
• At least 12 months experience working with an Object Oriented language
• Have C++ or Java knowledge:
  ?Creating Classes
  ?Inheritance and Abstraction
  ?Polymorphism
  ?Interfaces
  ?Exceptions
• Knowledge of the Visual Studio IDE.

Target Audience
This course is intended for experienced developers who already have programming experience in C, C++, Visual Basic, or Java and understand the concepts of object-oriented programming. This course is not designed for new programmers; it is targeted at professional developers with at least 12 months experience of programming in an object-oriented environment.

Course Objectives
After completing this course, students will be able to:
- Explain the purpose of the .NET Framework, and understand how to use C# and Visual Studio 2010 to build .NET Framework applications.
- Understand the syntax of basic C# programming constructs.
- Create and call methods in a C# application.
- Catch, handle and throw exceptions.
- Perform basic file I/O operations in a C# application.
- Create and use new types (enumerations, classes, and structures), and understand the differences between reference types and value types.
- Control the visibility and lifetime of members in a type.
- Use inheritance to create new reference types.
- Manage the lifetime of objects and control the use of resources.
- Define properties and indexers to encapsulate data, and define operators.
for this data.
- Decouple an operation from the method that implements an operation, and use these decoupled operations to handle asynchronous events.
- Use collections to aggregate data, and use Generics to implement type-safe collection classes, structures, interfaces, and methods.
- Implement custom collection classes that support enumeration.
- Query in-memory data by using LINQ.
- Integrate code written by using a dynamic language such as Ruby and Python, or technologies such as COM, into a C# application.

Course Outline

Introducing C# and the .NET Framework

Introduction to the .NET Framework
Creating Projects Within Visual Studio 2010
Writing a C# Application
Building a Graphical Application
Documenting an Application
Running and Debugging Applications by Using Visual Studio 2010
Lab: Introducing C# and the .NET Framework

Using C# Programming Constructs

Declaring Variables and Assigning Values
Using Expressions and Operators
Creating and Using Arrays
Using Decision Statements
Using Iteration Statements
Lab: Using C# Programming Constructs

Declaring and Calling Methods

Defining and Invoking Methods
Specifying Optional Parameters and Output Parameters
Lab: Declaring and Calling Methods

Handling Exceptions

Handling Exceptions
Raising Exceptions
Lab: Handling Exceptions

Reading and Writing Files

Accessing the File System
Reading and Writing Files by Using Streams
Lab: Reading and Writing Files
Creating New Types
Creating and Using Enumerations
Creating and Using Classes
Creating and Using Structs
Comparing References to Values
Lab : Creating New Types

Encapsulating Data and Methods
Controlling Visibility of Type Members
Sharing Methods and Data
Lab : Encapsulating Data and Methods

Inheriting From Classes and Implementing Interfaces
Using Inheritance to Define New Reference Types
Defining and Implementing Interfaces
Defining Abstract Classes
Lab : Inheriting From Classes and Implementing Interfaces

Managing the Lifetime of Objects and Controlling Resources
Introduction to Garbage Collection
Managing Resources
Lab : Managing the Lifetime of Objects and Controlling Resources

Encapsulating Data and Defining Overloaded Operators
Creating and Using Properties
Creating and Using Indexers
Overloading Operators
Lab : Creating and Using Properties
Lab : Creating and Using Indexers
Lab : Overloading Operators

Decoupling Methods and Handling Events
Declaring and Using Delegates
Using Lambda Expressions
Handling Events
Lab : Decoupling Methods and Handling Events
Using Collections and Building Generic Types

Using Collections
Creating and Using Generic Types
Defining Generic Interfaces and Understanding Variance
Using Generic Methods and Delegates
Lab : Using Collections
Lab : Building Generic Types

Building and Enumerating Custom Collection Classes

Implementing a Custom Collection Class
Adding an Enumerator to a Custom Collection Class
Lab : Building and Enumerating Custom Collection Classes

Using LINQ to Query Data

Using the LINQ Extension Methods and Query Operators
Building Dynamic LINQ Queries and Expressions
Lab : Using LINQ to Query Data

Integrating Visual C# Code with Dynamic Languages and COM Components

Integrating C# Code with Ruby and Python
Accessing COM Components from C#
Lab : Integrating C# Code with Dynamic Languages and COM Components