



COMPUTER TRAINING

The key to a better future

VISUAL STUDIO

Course No. 10266

5 Days

Target Audience:

This course teaches students C# language syntax, program structure, and implementation by using Microsoft Visual Studio 2010 and the Microsoft .NET Framework 4.0. It 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.

Pre-requisites:

Before attending this course, students must have the following pre-requisites:

- At least 12 months experience working with an Object Oriented language.
- Experience using C++, Java or another programming language.
- Knowledge of the Visual Studio integrated development environment (IDE).

Purpose:

After completing the course, students will be able 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 IO operations in a C# application; understand the difference 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; decouple an operation from the implementation method; use decoupled operations to handle asynchronous events; use collections to aggregate data; use Generics to implement type-safe collection classes, structures, interfaces, and methods; implement custom collection classes; query in-memory data by using LINQ; integrate code written by using a dynamic language or technology into a C# application.

Programming with C# with Microsoft Visual Studio 2010

At the end of the course the delegate will be able to complete the following:

Introducing C# and the .NET Framework

- This module explains the .NET Framework, and using C# and Visual Studio 2010 for building .NET Framework applications.
- Introduction to the .NET Framework 4.0
- Creating Projects within Visual Studio 2010
- Writing a C# Application
- Building a Graphical Application
- Documenting an Application
- Debugging Applications by Using Visual Studio 2010
- Lab: Introducing C# and the .NET Framework

Skills:

- Explain the purpose of the .NET Framework.
- Create Microsoft Visual C# projects by using Visual Studio 2010.
- Explain the structure of a C# application.
- Use the WPF Application template to build a simple graphical application.
- Use XML comments to document an application.
- Use the debugger to step through a program.

Using C# Programming Constructs

- This module explains the syntax of basic 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

Skills:

- Declare variables and assign values.
- Create expressions by using operators.
- Create and use arrays.
- Use decision statements.
- Use iteration statements.

Declaring and Calling Methods

- This module introduces the concept of methods and shows students how to declare and call methods using C#.
- Defining and Invoking Methods
- Specifying Optional Parameters and Output Parameters
- Lab: Declaring and Calling Methods

Skills:

- Describe how to declare and call methods.
- Define and call methods that take optional parameters and output parameters.

Handling Exceptions

- This module introduces the importance of exception handling and explains why applications should be designed with exception handling in mind.
- Handling Exceptions
- Raising Exceptions
- Lab: Handling Exceptions

Skills:

- Describe how to catch and handle exceptions.
- Describe how to create and raise exceptions.

Reading and Writing Files

- This module explains to read and write to files using the classes in the .NET Framework.
- Accessing the File System
- Reading and Writing Files by Using Streams
- Lab: Reading and Writing Files

Skills:

- Describe how to access the file system by using the classes that the .NET Framework provides.
- Describe how to read and write files by using streams.

Creating New Types

- This module explains how students can build their own types that model items in the real world, and explains the differences between reference types and value types.
- Creating and Using Enumerations
- Creating and Using Classes
- Creating and Using Structures
- Comparing References to Values
- Lab: Creating New Types

Skills:

- Describe how to create and use enumerations.
- Describe how to create and use classes.
- Describe how to create and use structures.
- Explain the differences between reference and value types.

Encapsulating Data and Methods

- This module describes how to use the access modifiers that C# provides to enable students to implement encapsulation.
- Controlling Visibility of Type Members
- Sharing Methods and Data
- Lab: Encapsulating Data and Methods

Skills:

- Describe how to control the visibility of type members.
- Describe how to share methods and data.

Inheriting from Classes and Implementing Interfaces

- This module describes how students can use inheritance, interfaces, and abstract classes to develop object hierarchies.
- Using Inheritance to Define New Reference Types
- Defining and Implementing Interfaces
- Defining Abstract Classes
- Lab: Inheriting from Classes and Implementing Interfaces

Skills:

- Use inheritance to define new reference types.
- Define and implement interfaces.
- Define abstract classes.



Microsoft Partner
Gold Learning

Managing the Lifetime of Objects and Controlling Resources

- This module introduces the concept of resource management and discusses its importance.
- Introduction to Garbage Collection
- Managing Resources
- Lab: Managing the Lifetime of Objects and Controlling Resources

Skills:

- Describe how garbage collection works in the .NET Framework.
- Manage resources effectively in an application.

Encapsulating Data and Defining Overloaded Operators

- This module explains how to create properties and indexers to encapsulate data, and how to define operators for this data.
- Creating and Using Properties
- Creating and Using Indexers
- Overloading Operators
- Lab: Creating and Using Properties
- Lab: Creating and Using Indexers
- Lab: Overloading Operators

Skills:

- Explain how properties work and use them to encapsulate data.
- Describe how to use indexers to access data through an array-like syntax.
- Describe how to use operator overloading to define operators for types.

Decoupling Methods and Handling Events

- This module explains how to decouple an operation from the method that implements it and how to use these decoupled methods to handle asynchronous events.
- Declaring and Using Delegates
- Using Lambda Expressions
- Handling Events
- Lab: Decoupling Methods and Handling Events

Skills:

- Describe the purpose of delegates, and explain how to use a delegate to decouple an operation from the implementing method.
- Explain the purpose of lambda expressions, and describe how to use a lambda expression to define an anonymous method.
- Explain the purpose of events, and describe how to use events to report that something significant has happened in a type that other parts of the application need to be aware of.

Using Collections and Building Generic Types

- This module introduces collections, and describes how to use Generics to implement type-safe collection classes, structures, interfaces, and methods.
- 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

Skills:

- Use collection classes.
- Define and use generic types.
- Define generic interfaces and explain the concepts of covariance and contra variance.
- Define and use generic methods and delegates.

Building and Enumerating Custom Collection Classes

- This module explains how to implement custom collection classes that support enumeration.
- Implementing a Custom Collection Class
- Adding an Enumerator to a Custom Collection Class
- Lab: Building and Enumerating Custom Collection Classes

Skills:

- Implement a custom collection class.
- Define an enumerator for a custom collection class.

Using LINQ to Query Data

- This module explains how students can use LINQ to abstract the mechanism that an application uses to query data from the application code.
- Using the LINQ Extension Methods and Query Operators
- Building Dynamic LINQ Queries and Expressions
- Lab: Using LINQ to Query Data

Skills:

- Describe how to use the LINQ extension methods and query operators.
- Describe how to build dynamic LINQ queries and expressions.

Integrating Visual C# Code with Dynamic Languages and COM Components

- This module explains how to integrate code written by using a dynamic language such as Ruby and Python, and technologies such as COM, into a C# application.
- Integrating Visual C# Code with Ruby and Python
- Accessing COM Components from Visual C#
- Lab: Integrating Visual C# Code with Dynamic Languages and COM Components

Skills:

- Integrate Ruby and Python code into a Visual C# application.
- Invoke COM components and services from a C# application.



F1 COMPUTER TRAINING SERVICES

THE KEY TO A BETTER FUTURE

for further information...

call us on
0800 169 1890

F1 COMPUTING SYSTEMS LTD

3 Kelso Place
Upper Bristol Road
BATH BA1 3AU

Fax: 01225 444728

training@f1comp.co.uk

www.f1comp.co.uk

LONDON BATH OR ONSITE

PARTICULARS

Cost:

For the latest available price, please see the course page on **www.f1comp.co.uk**, or call 01225 336096.

Numbers:

Maximum of 6 people on each course at F1's training facilities in London and Bath