



Complete Computing, Inc.

SERVING OUR CUSTOMERS SINCE 1982

20483BC: Programming in C#

Course length: 5 day(s)

Course Description

The goal of this course is to help students gain essential C# programming skills. This course is an entry point into the Windows Store apps training path. The course focuses on program structure, programming logic, defining and using variables and data types, implementing looping and branching, UI development with XAML, capturing input, storing data, basics of application lifecycle, handling exceptions, unit testing, etc., all within an object oriented programming approach to software development. The intended student customer for this training is a developer who has at least six months of professional experience. This student is expected to have limited exposure to C# coding. Students choosing to attend this training without professional software development experience should pay special attention to the training prerequisites. Developers who have more than 5 years programming experience may find that portions of this training are fundamental in nature when presenting the syntax associated with certain programming tasks. The lab scenario in this training was selected to support and demonstrate the structure for a variety of application scenarios. Although Windows Desktop applications will be represented within the lab activities, this training course is not designed to teach WPF application development at a professional level, but rather these solutions are used for context. The chosen lab scenario is intended to maintain a focus on the principals and coding components/structures/technologies that are used to establish a software application. This course maps to the 70-483 exam.

This training course teaches developers the programming skills that are required for developers to create Windows applications using the C# language. During their five days in the classroom students review the basics of C# program structure, language syntax, and implementation details, and then consolidate their knowledge throughout the week as they build an application that incorporates several features of the .NET Framework 4.5.

The course introduces many of the techniques and technologies employed by modern desktop and enterprise applications, including:

- Building new data types.
- Handling events.
- Programming the user interface.
- Accessing a database.
- Using remote data.
- Performing operations asynchronously.
- Integrating with unmanaged code.
- Creating custom attributes.
- Encrypting and decrypting data.

At the end of the course, students should leave the class with a solid knowledge of C# and how to use it to develop .NET Framework 4.5 applications.

400 WEST 7TH STREET (7TH & SPRING) LITTLE ROCK AR 72201-4288
PHONE: 501.372.3379 OR 800.880.2949 EMAIL: training@complete.com



Complete Computing, Inc.

SERVING OUR CUSTOMERS SINCE 1982

Audience Profile

This course is intended for experienced developers who already have programming experience in C, C++, JavaScript, Objective-C, Microsoft Visual Basic®, or Java and understand the concepts of object-oriented programming.

This course is not designed for students who are new to programming; it is targeted at professional developers with at least one month of experience programming in an object-oriented environment.

Prerequisites

Developers attending this course should already have gained some limited experience using C# to complete basic programming tasks. More specifically, students should have hands-on experience using C# that demonstrates their understanding of the following:

- How to name, declare, initialize and assign values to variables within an application.
- How to use:
 - arithmetic operators to perform arithmetic calculations involving one or more variables;
 - relational operators to test the relationship between two variables or expressions;
 - logical operators to combine expressions that contain relational operators.
- How to create the code syntax for simple programming statements using C# language keywords and recognize syntax errors using the Visual Studio IDE.
- How to create a simple branching structure using an IF statement.
- How to create a simple looping structure using a For statement to iterate through a data array.
- How to use the Visual Studio IDE to locate simple logic errors.
- How to create a Function that accepts arguments (parameters and returns a value of a specified type).
- How to design and build a simple user interface using standard controls from the Visual Studio toolbox.
- How to connect to a SQL Server database and the basics of how to retrieve and store data.
- How to sort data in a loop.
- How to recognize the classes and methods used in a program.



Complete Computing, Inc.

SERVING OUR CUSTOMERS SINCE 1982

Course Content

Module 1: Review of C# Syntax

This module reviews the core syntax and features of the C# programming language. It also provides an introduction to the Visual Studio 2012 debugger.

- **Overview of Writing Applications using C#**
- **Datatypes, Operators, and Expressions**
- **C# Programming Language Constructs**

Module 2: Creating Methods, Handling Exceptions, and Monitoring Applications

This module explains how to create and call methods, catch and handle exceptions. This module also describes the monitoring requirements of large-scale applications.

- **Creating and Invoking Methods**
- **Creating Overloaded Methods and Using Optional and Output Parameters**
- **Handling Exceptions**
- **Monitoring Applications**

Module 3: Developing the Code for a Graphical Application

This module describes how to implement the basic structure and essential elements of a typical desktop application, including using structures and enumerations, collections, and events.

- **Implementing Structs and Enums**
- **Organizing Data into Collections**
- **Handling Events**

Module 4: Creating Classes and Implementing Type-safe Collections

This module explains how to create classes, define and implement interfaces, and create and use generic collections. This module also describes the differences between value types and reference types in C#.

- **Creating Classes**
- **Defining and Implementing Interfaces**
- **Implementing Type-safe Collections**

Module 5: Creating a Class Hierarchy by Using Inheritance

This module explains how to use inheritance to create a class hierarchy and extend a .NET Framework class. This module also describes how to create generic classes and define extension methods.

- **Creating Class Hierarchies**
- **Extending .NET Framework Classes**
- **Creating Generic Types**

Module 6: Reading and Writing Local Data

This module explains how to read and write data by using file input/output (I/O) and streams, and how to serialize and deserialize data in different formats.

- **Reading and Writing Files**
- **Serializing and Deserializing Data**
- **Performing I/O Using Streams**

400 WEST 7TH STREET (7TH & SPRING) LITTLE ROCK AR 72201-4288
PHONE: 501.372.3379 OR 800.880.2949 EMAIL: training@complete.com



Complete Computing, Inc.

SERVING OUR CUSTOMERS SINCE 1982

Module 7: Accessing a Database

This module explains how to create and use an entity data model for accessing a database, and how to use LINQ to query and update data.

- **Creating and Using Entity Data Models**
- **Querying Data by Using LINQ**
- **Updating Data by Using LINQ**

Module 8: Accessing Remote Data

This module explains how to use the types in the System.Net namespace, and WCF Data Services, to query and modify remote data.

- **Accessing Data Across the Web**
- **Accessing Data in the Cloud**

Module 9: Designing the User Interface for a Graphical Application

This module explains how to build and style a graphical user interface by using XAML. This module also describes how to display data in a user interface by using data binding.

- **Using XAML to Design a User Interface**
- **Binding Controls to Data**
- **Styling a User Interface**

Module 10: Improving Application Performance and Responsiveness

This module explains how to improve the throughput and response time of applications by using tasks and asynchronous operations.

- **Implementing Multitasking by using Tasks and Lambda Expressions**
- **Performing Operations Asynchronously**
- **Synchronizing Concurrent Access to Data**

Module 11: Integrating with Unmanaged Code

This module explains how to integrate unmanaged libraries and dynamic components into a C# application. This module also describes how to control the lifetime of unmanaged resources.

- **Creating and Using Dynamic Objects**
- **Managing the Lifetime of Objects and Controlling Unmanaged Resources**

Module 12: Creating Reusable Types and Assemblies

This module explains how to examine the metadata of types by using reflection, create and use custom attributes, generate managed code at runtime, and manage different versions of assemblies.

- **Examining Object Metadata**
- **Creating and Using Custom Attributes**
- **Generating Managed Code**
- **Versioning, Signing and Deploying Assemblies**



Complete Computing, Inc.

SERVING OUR CUSTOMERS SINCE 1982

Module 13: Encrypting and Decrypting Data

This module explains how to encrypt and decrypt data by using symmetric and asymmetric encryption.

- **Implementing Symmetric Encryption**
- **Implementing Asymmetric Encryption**