

Course Details

Course 20483A Programming in C#

Length: 5 Days

Published: January 14, 2013

Language(s): English

Audience(s): Developers.

Level: 200

Technology: Microsoft Visual Studio 2012

Type: Course

Delivery Method: Instructor-led (classroom)

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.

Course Outline

Module 1: Review of C# Syntax

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

Module 2: Creating Methods, Handling Exceptions, and Monitoring 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

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

Module 5: Creating a Class Hierarchy by Using Inheritance

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

Module 7: Accessing a Database

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

Module 9: Designing the User Interface for a Graphical Application

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

Module 11: Integrating with Unmanaged Code

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

Module 13: Encrypting and Decrypting Data

- Implementing Symmetric Encryption
- Implementing Asymmetric Encryption

Module 4: Creating Classes and Implementing Type-safe Collections

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

Module 6: Reading and Writing Local Data

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

Module 8: Accessing Remote Data

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

Module 10: Improving Application Performance and Responsiveness

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

Module 12: Creating Reusable Types and Assemblies

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