



20483 Programming in C#

This training course teaches developers the programming skills that are required for developers to create Windows applications using the C# language

Length Days: 5 | Length Hours: 40

Target Audience

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.

Course Objectives

After completing this course, students will be able to:

- Describe the core syntax and features of C#.
- Create and call methods, catch and handle exceptions, and describe the monitoring requirements of large-scale applications.
- Implement the basic structure and essential elements of a typical desktop application.
- Create classes, define and implement interfaces, and create and use generic collections.
- Use inheritance to create a class hierarchy, extend a .NET Framework class, and create generic classes and methods.
- Read and write data by using file input/output and streams, and serialize and deserialize data in different formats.
- Create and use an entity data model for accessing a database and use LINQ to query and update data.
- Use the types in the System.Net namespace and WCF Data Services to access and query remote data.
- Build a graphical user interface by using XAML.
- Improve the throughput and response time of applications by using tasks and asynchronous operations.
- Integrate unmanaged libraries and dynamic components into a C# application.
- Examine the metadata of types by using reflection, create and use custom attributes, generate code at runtime, and manage assembly versions.
- Encrypt and decrypt data by using symmetric and asymmetric encryption.

Course Outline

1 - Review of C# Syntax

- Overview of Writing Applications using C#
- Datatypes, Operators, and Expressions
- C# Programming Language Constructs
- Lab : Developing the Class Enrolment Application

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
- Lab : Extending the Class Enrolment Application Functionality

3 - Developing the Code for a Graphical Application

- Implementing Structs and Enums
- Organizing Data into Collections
- Handling Events
- Lab : Writing the Code for the Grades Prototype Application

4 - Creating Classes and Implementing Type-safe Collections

- Creating Classes
- Defining and Implementing Interfaces
- Implementing Type-safe Collections
- Lab : Adding Data Validation and Type-safety to the Grades Application

5 - Creating a Class Hierarchy by Using Inheritance

- Creating Class Hierarchies
- Extending .NET Framework Classes
- Creating Generic Types
- Lab : Refactoring Common Functionality into the User Class

6 - Reading and Writing Local Data

- Reading and Writing Files
- Serializing and Deserializing Data
- Performing I/O Using Streams
- Lab : Generating the Grades Report

7 - Accessing a Database

- Creating and Using Entity Data Models
- Querying Data by Using LINQ
- Updating Data by Using LINQ
- Lab : Retrieving and Modifying Grade Data

8 - Accessing Remote Data

- Accessing Data Across the Web
- Accessing Data in the Cloud
- Lab : Retrieving and Modifying Grade Data in the Cloud

9 - Designing the User Interface for a Graphical Application

- Using XAML to Design a User Interface
- Binding Controls to Data
- Styling a User Interface
- Lab : Customizing Student Photographs and Styling the Application

10 - Improving Application Performance and Responsiveness

- Implementing Multitasking by using Tasks and Lambda Expressions
- Performing Operations Asynchronously
- Synchronizing Concurrent Access to Data
- Lab : Improving the Responsiveness and Performance of the Application

11 - Integrating with Unmanaged Code

- Creating and Using Dynamic Objects
- Managing the Lifetime of Objects and Controlling Unmanaged Resources
- Lab : Upgrading the Grades Report

12 - Creating Reusable Types and Assemblies

- Examining Object Metadata
- Creating and Using Custom Attributes
- Generating Managed Code
- Versioning, Signing and Deploying Assemblies
- Lab : Specifying the Data to Include in the Grades Report

13 - Encrypting and Decrypting Data

- Implementing Symmetric Encryption
- Implementing Asymmetric Encryption
- Lab : Encrypting and Decrypting Grades Reports