

Programming in C# - Windows Applications

Description

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

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.

Delegates will learn how to

- about 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.
-

Outline

Review of C# Syntax

Creating Methods, Handling Exceptions, and Monitoring Applications

Developing the Code for a Graphical Application

Creating Classes and Implementing Type-safe Collections

Creating a Class Hierarchy by Using Inheritance

Reading and Writing Local Data

Accessing a Database

Accessing Remote Data

Designing the User Interface for a Graphical Application

Improving Application Performance and Responsiveness

Integrating with Unmanaged Code

Creating Reusable Types and Assemblies

Encrypting and Decrypting Data