20483C Programming in C#

Register Online

Schedule

Class Length: 5 Days

G2R = “Guaranteed to Run” | OLL = “Online LIVE”
ILT = “Instructor-Led-Training”

This course is not currently available on the public schedule. Please contact us using the information in the footer below to inquire about future dates or to schedule a private class.

Overview

This course teaches developers the programming skills that are required for developers to create Windows applications using the Visual C# language. Students will review the basics of Visual 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 4.7.

Prerequisite Comments

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#.

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

"• Describe the core syntax and features of Visual C#.
• Create methods, 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 and to extend a .NET Framework class.
Read and write data by using file input/output and streams, and serialize and de-serialize data in different formats.
Create and use an entity data model for accessing a database and use LINQ to query data.
Access and query remote data by using the types in the System.Net namespace and WCF Data Services.

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 Visual 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 Application by Using Visual C#
Data Types, Operators, and Expressions
Visual C# Programming Language Constructs

Lab : Implementing Edit Functionality for the Students List
Implementing Insert Functionality for the Students List
Implementing Delete Functionality for the Students List
Displaying a Student's Age

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

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
Refactoring the Enrolment Code
Validating Student Information
Saving Changes to the Class List

3 - Basic types and constructs of Visual C#

Implementing Structs and Enums
Organizing Data into Collections
Handling Events

Lab : Writing the Code for the Grades Prototype Application
Adding Navigation Logic to the Grades Prototype Application
Creating Data Types to Store User and Grade Information
Displaying User and Grade Information

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

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 Application
Implementing the Teacher, Student, and Grade Structs as Classes
Adding Data Validation to the Grade Class
Displaying Students in Name Order
Enabling Teachers to Modify Class and Grade Data

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

5 - Creating a Class Hierarchy by Using Inheritance

Creating Class Hierarchies
Extending .NET Framework Classes

Lab : Refactoring Common Functionality into the User Class
Refactoring Common Functionality into the User Class
Implementing Password Complexity by Using an Abstract Method
Creating the ClassFullException Custom Exception

6 - Reading and Writing Local Data

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

6 - Reading and Writing Local Data

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

Lab : Generating the Grades Report
Serializing Data for the Grades Report as XML
Previewing the Grades Report
Persisting the Serialized Grade Data to a File

7 - Accessing a Database

Creating and Using Entity Data Models
Querying Data by Using LINQ

Lab : Retrieving and Modifying Grade Data
Creating an Entity Data Model from The School of Fine Arts Database
Updating Student and Grade Data by Using the Entity Framework
Extending the Entity Data Model to Validate Data

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

8 - Accessing Remote Data

Accessing Data Across the Web
Accessing Data by Using OData Connected Services

Lab : Retrieving and Modifying Grade Data Remotely
Creating a WCF Data Service for the SchoolGrades Database
Integrating the Data Service into the Application
Retrieving Student Photographs Over the Web

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

9 - Designing the User Interface for a Graphical Application

Using XAML to Design a User Interface
Binding Controls to Data

Lab : Customizing Student Photographs and Styling the Application
Customizing the Appearance of Student Photographs
Styling the Logon View
Animating the StudentPhoto Control

10 - Improving Application Performance and Responsiveness

Implementing Multitasking
Performing Operations Asynchronously
Synchronizing Concurrent Access to Data

Lab : Improving the Responsiveness and Performance of the Application
Ensuring That the UI Remains Responsive When Retrieving Teacher Data
Providing Visual Feedback During Long-Running Operations

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

11 - Integrating with Unmanaged Code

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

Lab : Upgrading the Grades Report
Generating the Grades Report by Using Word
Controlling the Lifetime of Word Objects by Implementing the Dispose Pattern

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

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
Creating and Applying the IncludeInReport attribute
Updating the Report
Storing the Grades.Utilities Assembly Centrally

13 - Encrypting and Decrypting Data

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

13 - Encrypting and Decrypting Data

Implementing Symmetric Encryption
Implementing Asymmetric Encryption

Lab : Encrypting and Decrypting the Grades Report
Encrypting the Grades Report
Encrypting the Grades Report

Related Courses, Certifications, Exams

- Exam 70-483 - Programming in C#