TRENDS Academy
Learning | Evolving | Empowering

COURSE OUTLINE

Course 5539AC: Programming in C#

Duration: 5 Days

Course Prerequisite:

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 in data in a loop.
- How to recognize the classes and methods used in a program.

Course Description:

This five-day 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 6.0.

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 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 of inheritance to create a class hierarchy and to extend a .NET class.
- 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 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 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:

Module 1: Review of C# Syntax

Lessons:
- ➢ Overview of Writing Application by Using C#
- ➢ Data Types, Operators, and Expressions
- ➢ C# Programming Language Constructs

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

Lessons:
- ➢ Creating and Invoking Methods
- ➢ Creating Overloaded Methods and Using Optional and Output Parameters
- ➢ Handling Exceptions
- ➢ Monitoring Applications

Module 3: Basic types and constructs of C#

Lessons:
- ➢ Implementing Structs and Enums
- ➢ Organizing Data into Collections
- ➢ Handling Events

Module 4: Creating Classes and Implementing Type-safe Collections

Lessons:
- ➢ Creating Classes
- ➢ Defining and Implementing Interfaces
- ➢ Implementing Type-Safe Collections

Module 5: Creating a Class Hierarchy by Using Inheritance

Lessons:
- ➢ Creating Class Hierarchies
- ➢ Extending .NET Classes

Module 6: Reading and Writing Local Data

Lessons:
- ➢ Reading and Writing Files
- ➢ Serializing and Deserializing Data
- ➢ Performing I/O by Using Streams

Module 7: Accessing a Database

Lessons:
- ➢ Creating and Using Entity Data Models
- ➢ Querying Data by Using LINQ

Module 8: Accessing Remote Data

Lessons:
- ➢ Accessing Data Across the Web
- ➢ Accessing Data by Using OData Connected Services

Module 9: Designing the User Interface for a Graphical Application

Lessons:
- ➢ Using XAML to Design a User Interface
- ➢ Binding Controls to Data

Module 10: Improving Application Performance and Responsiveness

Lessons:
- ➢ Implementing Multitasking
- ➢ Performing Operations Asynchronously
- ➢ Synchronizing Concurrent Access to Data

Module 11: Integrating with Unmanaged Code

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

Module 12: Creating Reusable Types and Assemblies

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

Module 13: Securing Data

Lessons:
- ➢ Implementing Symmetric Encryption
- ➢ Implementing Asymmetric Encryption

REGISTER NOW!

training@trends.com.ph
(+632) 8863-2123
www.trendsacademy.com.ph