Document Generated: 04/11/2026

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 5 Days

Next Course Date: June 15, 2026

Programming in C# - Creating Applications in C# and .Net Core (TTCN20483)

About This Course:

Programming in C# / Creating Apps in C# and .NET Core is a hands-on, expert-led course designed to help you become confident and capable building real-world applications using C# and the modern .NET Core platform. C# is used everywhere from enterprise software to cloud services and desktop apps, and this course gives you the practical skills you need to write clean, efficient, and well-structured code that supports performance, reliability, and long-term growth. With a focus on doing rather than just watching, you will gain the technical know-how to apply these skills where they matter most.

Throughout the course, you will build your skills in a way that connects directly to real development work. You will learn how to structure code using classes, interfaces, and methods so your applications are easier to read, test, and maintain. You will practice handling errors gracefully, working with files and databases, and designing applications that stay responsive and reliable. By focusing on modern patterns like async programming and learning to access data both locally and remotely, you will be better prepared to solve common development challenges in a

thoughtful and scalable way.

This course is for developers who need to learn or strengthen their C# programming skills. Whether you are moving into a new role, joining a project that uses .NET, or updating your skills for modern development practices, you will benefit from a supportive environment where an expert instructor leads you through each concept step by step. With about 50 percent of the class dedicated to hands-on labs, you will get plenty of practice applying what you learn so you walk away ready to build useful, reliable applications with confidence.

Course Objectives:

- Write clean, structured C# code using modern syntax and tools. Learn how to use top-level statements, global usings, and other updated features to keep your code simple, readable, and easy to manage.
- Build reusable code with methods, classes, and interfaces. Understand how to organize logic into meaningful parts that make your applications easier to develop, test, and maintain over time.
- Handle errors properly and keep your applications stable. Gain experience using exception handling, logging, and tracing so you can build apps that behave reliably in the real world.
- Work with data using files, streams, and databases. Practice reading and writing data from files, using serialization, and accessing databases with Entity Framework and LINQ to build data-driven applications.
- Create responsive apps that stay fast and smooth. Use tasks, async programming, and threading to keep your applications responsive even during long operations or data processing.
- Design modern user interfaces and connect to remote data. Explore simple ways to build interfaces using MAUI, Blazor, or ASP.NET and learn how to send and receive data across the web for more connected experiences.

Audience:

Developers attending this course should have incoming 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 recognize the classes and methods used in a program.

Prerequisites:

Developers attending this course should have incoming 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 recognize the classes and methods used in a program.

Course Outline:

Chapter 1: Review of C# Syntax & New Features

Get started writing clean, modern C# code with updated features and the tools you will use throughout the course.

- Understand Visual Studio and project setup
- Compare .NET Core and .NET Framework
- Create simple console applications
- Use top-level statements effectively
- Apply global usings to clean code
- Explore core C# building blocks

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

Learn how to organize your code, manage errors gracefully, and keep applications running smoothly using real-world techniques.

- Write and call reusable methods
- Create overloaded methods and optional parameters
- Handle exceptions to improve stability
- Monitor applications using logs and tracing
- Track errors with real-world techniques
- Build smarter, more reliable applications

Chapter 3: Basic Types and Constructs of C#

Build a solid foundation by working with core data types, collections, and events to structure your application logic.

- Create and use structs and enums
- Organize data with collections
- Store and retrieve items easily
- Loop through data the right way
- Subscribe to and handle events
- Make code flexible with core types

Chapter 4: Creating Classes and Implementing Type-Safe Collections

Learn how to design custom types that are easy to reuse, maintain, and apply to real projects.

- Define and create classes
- Build and use interfaces
- Organize code for reusability
- Implement generic collections safely
- Make types easy to maintain
- Structure applications the smart way

Chapter 5: Creating a Class Hierarchy by Using Inheritance

Use inheritance to build connected class structures that let you reuse logic and extend your applications easily.

- Create base and derived classes
- Extend functionality through inheritance
- Reuse code across related types
- Simplify changes with structured hierarchy
- Improve application flexibility with design
- Practice real-world class structures

Chapter 6: Reading and Writing Local Data

Work with local files and data streams to store, read, and manage information efficiently within your applications.

- Read and write files with C#
- Serialize and deserialize application data
- Use streams for large data handling
- Have and load data easily
- Work with different file formats
- Connect applications to local storage

Chapter 7: Accessing a Database

Learn how to connect to a database, retrieve and update data, and build useful data-driven features.

- Set up Entity Framework connections
- Query databases using LINQ
- Manage data through C# code
- Build data-driven application features
- Perform basic create, read, update, delete
- Explore real-world database scenarios

Chapter 8: Accessing Remote Data

Send and receive information across the web to build more connected and useful applications.

- Send requests to web services
- Receive and process remote data
- Connect applications to external APIs
- Work with online data sources
- Build lightweight web-connected apps
- Handle remote data safely

Chapter 9: Designing the User Interface for a Graphical Application

Explore different ways to build user interfaces for desktop or web apps that look good and respond to users.

- Create MAUI apps for cross-platform use
- Build simple UIs with Blazor
- Develop ASP.NET web applications
- Understand UI layout basics
- Connect user actions to functionality
- Make user-friendly and responsive designs

Chapter 10: Improving Application Performance and Responsiveness

Keep your apps fast and responsive by learning techniques that prevent delays and manage background work.

- Use tasks for multitasking operations
- Handle long-running processes smoothly
- Apply async and await correctly
- Coordinate work across multiple threads
- Improve app speed and responsiveness
- Build better user experiences

Chapter 11: Creating Reusable Types and Assemblies

Add flexibility to your applications by creating reusable code and working with advanced features like attributes and reflection.

- Inspect code with reflection
- Create and use custom attributes
- Work with object metadata
- Package reusable code cleanly
- Add flexible functionality to apps
- Build smarter, scalable applications