Programming in C# | Creating Apps in C# and .Net Core (TTCN20483)

Category: Software Development & Programming | **Vendor:** Trivera Tech

Duration: 40.00 hours (5 days) **32.5 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Classroom

Course Overview

C# Web Developer / Developing ASP.Net Core MVC Web Applications is an intensive, hands-on training course, essential for experienced developers seeking instruction in creating web applications in C# & .Net Core. Throughout the course web developers will learn to develop advanced ASP.NET Core MVC applications using .NET Core tools and technologies. The focus will be on coding activities that enhance the performance and scalability of the Web site application This independent course aligns with the topics and skills in Microsoft Official Curriculum (MOC) course 20486.

About This Course

C# Web Developer / Developing ASP.Net Core MVC Web Applications is an intensive, hands-on training course, essential for experienced developers seeking instruction in creating web applications in C# & .Net Core. Throughout the course web developers will learn to develop advanced ASP.NET Core MVC applications using .NET Core tools and technologies. The focus will be on coding activities that enhance the performance and scalability of the Web site application This independent course aligns with the topics and skills in Microsoft Official Curriculum (MOC) course 20486.

Who Should Attend

This course is intended for professional web developers who use Microsoft Visual Studio in an individual-based or team-based, small-sized to large development environment. Candidates for this course are interested in developing advanced web applications and want to manage the rendered HTML comprehensively. They want to create websites that separate the user interface, data access, and application logic.

Prerequisites & Entry Requirements

General 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.

Learning Outcomes

Upon successful completion of this course, participants will be able to:

Our engaging instructors and mentors are highly experienced practitioners who bring years of current 'on-the-job' experience into every classroom. Working in a hands-on learning environment, guided by our expert team, attendees will learn to:

Describe the Microsoft Web Technologies stack and select an appropriate technology to use to develop any given application.

Design the architecture and implementation of a web application that will meet a set of functional requirements, user interface requirements, and address business models.

Configure the pipeline of ASP.NET Core web applications using middleware, and leverage dependency injection across MVC application.

Add Controllers to an MVC Application to manage user interaction, update models, and select and return Views.

Develop a web application that uses the ASP.NET Core routing engine to present friendly URLs and a logical navigation hierarchy to users.

Create Views in an MVC application that display and edit data and interact with Models and Controllers.

Create MVC Models and write code that implements business logic within Model methods, properties, and events.

Connect an ASP.NET Core application to a database using Entity Framework Core.

Implement a consistent look and feel across an entire MVC web application.

Write JavaScript code that runs on the client-side and utilizes the jQuery script library to optimize the responsiveness of an MVC web application.

Add client side packages and configure Task Runners.

Run unit tests and debugging tools against a web application in Visual Studio 2017.

Write an MVC application that authenticates and authorizes users to access content securely using Identity.

Build an MVC application that resists malicious attacks.

Use caching to accelerate responses to user requests.

Use SignalR to enable two-way communication between client and server.

Describe what a Web API is and why developers might add a Web API to an application.

Describe how to package and deploy an ASP.NET Core MVC web application from a development computer to a web server.

Detailed Course Outline

Module 1: Exploring ASP.NET Core MVC

- Overview of Microsoft Web Technologies
- Overview of ASP.NET 4.x
- Introduction to ASP.NET Core MVC
-

Module 2: Designing ASP.NET Core MVC Web Applications

- Planning in the Project Design Phase
- Designing Models, Controllers and Views
-

Module 3: Configure Middlewares and Services in ASP.NET Core

- Configuring Middlewares
- Configuring Services

Module 4: Developing Controllers

- Writing Controllers and Actions
- Configuring Routes
- Writing Action Filters

Module 5: Developing Views

- Creating Views with Razor Syntax
- Using HTML Helpers and Tag Helpers
- Reusing Code in Views

Module 6: Developing Models

- Creating MVC Models
- Working with Forms
- Validate MVC Application

Module 7: Using Entity Framework Core in ASP.NET Core

- Introduction to Entity Framework Core
- Working with Entity Framework Core
- Use Entity Framework Core to connect to Microsoft SQL Server

Module 8: Using Layouts, CSS and JavaScript in ASP.NET Core MVC

- Using Layouts
- Using CSS and JavaScript
- Using jQuery

Module 9: Client-Side Development

- Applying Styles
- Using Task Runners
- Responsive design

Module 10: Testing and Troubleshooting

- Testing MVC Applications
- Implementing an Exception Handling Strategy
- Logging MVC Applications

Module 11: Managing Security

- Authentication in ASP.NET Core
- Authorization in ASP.NET Core
- Defending from Attacks

Module 12: Performance and Communication

- Implementing a Caching Strategy
- Managing State
- Two-way communication

Module 13: Implementing Web APIs

- Introducing Web APIs
- Developing a Web API
- Calling a Web API

Module 14: Hosting and Deployment

- On-premise hosting and deployment
- Deployment to Microsoft Azure
- Microsoft Azure Fundamentals

Module 15 Accessing Remote Data

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

Module 16: Designing the User Interface for a Graphical Application

- Using XAML to Design a User Interface
- Binding Controls to Data

Module 17: Improving Application Performance and Responsiveness

- Implementing Multitasking
- Performing Operations Asynchronously
- Synchronizing Concurrent Access to Data

Module 18: Creating Reusable Types and Assemblies

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

Module 19: Encrypting and Decrypting Data

- Implementing Symmetric Encryption
- Implementing Asymmetric Encryption

, Module 1: Review of Visual 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

Module 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

Module 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

Module 4: Creating Classes and Implementing Type-Safe Collections

Creating ClassesDefining and Implementing Interfaces

Implementing Type-Safe Collections

Lab: Adding Data Validation and Type-Safety to the Application

Module 5: Creating a Class Hierarchy by Using Inheritance

Creating Class Hierarchies

Extending .NET Framework Classes

Lab: Refactoring Common Functionality into the User Class

Module 6: Reading and Writing Local Data

Reading and Writing Files

Serializing and Deserializing Data

Performing I/O by Using Streams

Lab: Generating the Grades ReportModule 6: Reading and Writing Local Data

Module 7: Accessing a Database

Creating and Using Entity Data Models

Querying Data by Using LINQ

Lab: Retrieving and Modifying Grade Data

Module 8: Accessing Remote Data

Accessing Data Across the Web

Accessing Data by Using OData Connected Services

Lab: Retrieving and Modifying Grade Data Remotely

Module 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

Module 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

Module 11: Creating Reusable Types and Assemblies

Examining Object Metadata

Creating and Using Custom Attributes

Generating Managed Code

Versioning, and Deploying Assemblies

Lab: Specifying the Data to Include in the Grades Report

Module 12: Encrypting and Decrypting Data

Implementing Symmetric Encryption

Implementing Asymmetric Encryption

Lab: Encrypting and Decrypting the Grades Report

Additional Course Details

Nexus Humans Programming in C# | Creating Apps in C# and .Net Core (TTCN20483) training program is a workshop that presents an invigorating mix of sessions, lessons, and masterclasses meticulously crafted to propel your learning expedition forward.

This immersive bootcamp-style experience boasts interactive lectures, hands-on labs, and collaborative hackathons, all strategically designed to fortify fundamental concepts.

Guided by seasoned coaches, each session offers priceless insights and practical skills crucial for honing your expertise. Whether you're stepping into the realm of professional skills or a seasoned professional, this comprehensive course ensures you're equipped with the knowledge and prowess necessary for success.

While we feel this is the best course for the Programming in C# | Creating Apps in C# and .Net Core (TTCN20483) course and one of our Top 10 we encourage you to read the course outline to make sure it is the right content for you.

Additionally, private sessions, closed classes or dedicated events are available both live online and at our training centres in Dublin and London, as well as at your offices anywhere in the UK, Ireland or across EMEA.

Frequently Asked Questions

Q: What delivery options are available for Programming in C# | Creating Apps in C# and .Net Core (TTCN20483)?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 5-day Programming in C# | Creating Apps in C# and .Net Core (TTCN20483) course provides up to 32.5 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Programming in C# | Creating Apps in C# and .Net Core (TTCN20483) training?

The training takes place over 5 day(s), with each day lasting approximately 40.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Programming in C# | Creating Apps in C# and .Net Core (TTCN20483)?

Yes, we provide corporate training, dedicated training, and closed classes for Programming in C# | Creating Apps in C# and .Net Core (TTCN20483). Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: C# / .Net

Q: Why choose Nexus Human for Programming in C# | Creating Apps in C# and .Net Core (TTCN20483)?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your Programming in C# | Creating Apps in C# and .Net Core (TTCN20483) training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 13/11/2026.