ASP.NET MVC Programming Using C# and .NET6

Duration: 5 Days *(Face-to-Face & Remote-Live)*, or 35 Hours *(On-Demand)*

Price: $2495 *(Face-to-Face & Remote-Live)*, or $1495 *(On-Demand)*

Discounts: We offer multiple discount options. Click here for more information.

Delivery Options: Attend face-to-face in the classroom, remote-live or on-demand training.

Students Will Learn

- Using Visual Studio to create C# applications
- Working with .NET data types
- Creating variables with the proper scope and using operators to build complex expressions
- Designing and using classes
- Using control structures such as `if`, `while` and `for`
- Using procedures to build complex applications
- Throwing and trapping exceptions using the `try` and `catch` statements
- Using single and multi-dimensional arrays
- Working with .NET collections
- Using LINQ to make queries
- Defining and implementing interfaces
- Working with enumerations
- Architecture of ASP.NET MVC web applications
- Using Visual Studio's project templates to create specific types of MVC applications
- Using MVC's Dependency Injection container to create application-scoped services
- Creating controllers containing action methods to process HTTP requests
- Using the Razor view engine to design ASP.NET MVC views to render responses to HTTP requests
- Creating data model classes manually and by using the Entity Framework to populate strongly-typed views
- Creating custom ASP.NET MVC routes
- Using Visual Studio's tools to create and run tests for ASP.NET MVC applications
- Securing and deploying ASP.NET MVC web applications

Course Description

This course provides students with hands on experience using Visual Studio to create dynamic web applications using ASP.NET MVC, C#, and .NET6. The class provides a thorough introduction to the C# programming language, including coverage of the

essentials of the C# programming language, built in data types, operators, control structures, classes and methods, collections and exception handling.

Students learn how to leverage the power of the Model-View-Controller design pattern with the ASP.NET MVC design pattern to separate the layers of a web application. Students will use the Razor view engine to design a user interface, and will learn how to build models to manage an application's data layer using both the Entity Framework and LINQ. Students also learn how to build controllers containing action methods to manage communication between views and models.

Other topics include data scaffolding; URL routing; implementing security; unit testing; and deploying ASP.NET MVC applications. Comprehensive labs provide the students with experience creating, debugging, testing and deploying dynamic ASP.NET MVC applications.

This course provides thorough coverage of the use of **ASP.NET MVC** for creation of web applications. Students requiring additional coverage of **ASP.NET Web Forms**, **Windows Forms**, **WCF** or **Windows Presentation Foundation** should contact HOTT or refer to HOTT's complete course listing for additional training courses.

Students who are already familiar with C# language syntax and wish to learn about ASP.NET MVC applications in the context of .NET6 may want to take the 3-day ASP.NET MVC Programming for Experienced C# Programmers Using .NET6 course instead.

Students who are already familiar with C# language syntax and wish to learn about ASP.NET MVC applications in the context of .NET Frameworks 4.0 through 4.8 may want to take the 3-day ASP.NET MVC Programming for Experienced C# Programmers class instead.

Course Prerequisites

Knowledge of fundamental HTML syntax is helpful, but not required. Prior experience with a scripting or programming language is required.

Course Overview

Introduction to .NET

- Overview of the .NET Framework
- How .NET is Different from Traditional Programming
- Common Language Runtime (CLR)
- Common Language Specification (CLS)
- Common Type System (CTS)
- .NET Assemblies
- Microsoft Intermediate Language (CIL)
- .NET Namespaces
- .NET Framework Class Library

Introduction to Visual Studio

- Creating a Project
- Using the Code Editor
- Correcting Syntax Errors
- Setting Project Properties
- Adding References
- Compiling a Program
- Running a Program
- Debugging a Program
- Using the MSDN (Help)

Language Fundamentals

- C# Program Structure
- Defining Namespaces

Conditionals and Looping

- `if/else`
- `switch`

- Understanding C# Data Types
- Defining Variables and Constants
- Comparing Value Types vs. Reference Types
- Working with Operators and Expressions
- Performing Type Conversions
- Using Console I/O
- Formatting Numbers, Date and Times

- `while` and `do/while`
- `for`
- `foreach`

Methods and Parameters

- Defining Static and Instance Methods
- Passing Parameters by value and by reference
- Overloading Methods
- Using Variable Length Parameter Lists

Exception Handling

- What are Exceptions?
- .NET Exception Hierarchy
- Catching Exceptions
- Throwing Exceptions
- Managing Resources with Finally

Collections

- Defining and Using Arrays
- Understanding `System.Array`
- .NET Collections vs Generic Collections
- Working with Lists
- Working with Dictionaries
- Using LINQ to Objects

Object-Oriented Programming

- Overview of Object-Oriented Programming
- Building Classes
- Defining Properties
- Using Auto-Implemented Properties
- Defining Methods
- Understanding Constructors
- Extending .NET Classes via Inheritance
- Defining and Implementing Interfaces
- Understanding the Role of Interfaces in .NET

Overview of ASP.NET MVC

- Overview of Model-View-Controller Design Pattern
- ASP.NET MVC Application Architecture
- Understanding the MVC Execution Process
- Building an ASP.NET MVC Application Using Visual Studio
- Visual Studio MVC Project Templates
- Using a `web.config` File

Startup and Services

- Fundamentals of Dependency Injection
- Using the default DI container
- Using the Startup file
- Creating Services and Controlling Service Lifetime
- Building an ASP.NET MVC Application Using Visual Studio Project Templates
- Using Bootstrap CSS for the UI

Developing Views

- Creating Views
- Using the Razor View Engine
- Using ViewBag to Pass Controller Data
- Using HTML Helpers

Developing Models

- Creating Model Classes
- Working with Strongly-Typed Views
- Validating User Input
- Using Data Annotations for Display
- Working with the Entity Framework
- Working with LINQ to SQL
- Using Scaffolding to Generate Views

Developing Controllers

Routing Control

- Creating Controllers
- Defining Action Methods
- Working with HTML Form Data
- Mapping URLs to Action Methods
- Understanding `ActionResult` Types
- Using Model Binding

- Understanding Routing in ASP.NET MVC
- Defining URL Routes
- Registering Routes
- Adding Constraints to Routes
- Using Attribute Routing

MVC Unit Testing

- Test-Driven Development
- Designing Test Cases
- Creating Unit Tests
- Using xUnit for Testing
- Using Visual Studio Test Explorer

Securing MVC Applications

- ASP.NET Security
- Securing the Application with Individual Accounts
- Securing the Application with Windows Accounts
- Using Attributes to Leverage Access
- Defending against Attacks
    - Cross-site Scripting
    - Session Hijacking
    - SQL Injection
    - Input Forgery

Deploying ASP.NET MVC Applications

- Understanding the Kestrel Web Server
- Using Publish Profiles
- Choosing a Deployment Destination
- App Config using Environments
- Configuring IIS for ASP.NET MVC Deployment

Hands On Technology Transfer
The Best Way to Transfer Technology Skills

1 Village Square, Suite 8
14 Fletcher Street
Chelmsford, MA 01824