

## ASP.NET MVC Programming Using C#

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

**Price:** CDN\$3,275 (*Face-to-Face & Remote-Live*), or CDN\$1,995 (*On-Demand*)

**Discounts:** We offer multiple discount options. [Click here](#) for more info.

**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 to create Internet and Intranet applications
- Creating controllers containing action methods to process HTTP requests
- Using both the ASPX and Razor view engines to design ASP.NET MVC views to render responses to HTTP requests
- Creating data models by hand, by using typed datasets, and by using the Entity Framework
- Creating and debugging ASP.NET MVC routines
- Using MS Test and NUnit to create and run tests for ASP.NET MVC applications
- Integrating ASP.NET Web Forms and ASP.NET MVC within one web application
- 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 and C#. 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 Framework to separate the layers of a web application. Students will use the ASPX and Razor view engines to design a user interface. Students will learn how to build models to manage an application's data layer using both the Entity Framework and LINQ to SQL. And students will learn how to build controllers containing action methods to manage communication between views and models.

Other topics include data scaffolding; URL routing; implementing security; using MVC and Web Forms in the same application, 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 the C# language syntax may want to take the 3-day [ASP.NET MVC Programming for Experienced C# Programmers](#) course 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
- Understanding C# Data Types
- Defining Variables and Constants
- Comparing Value Types vs. Reference Types
- Working with Operators and Expressions

### Conditionals and Looping

- if/else
- switch
- while and do/while
- for
- foreach

- Performing Type Conversions
- Using Console I/O
- Formatting Numbers, Date and Times

## Methods and Parameters

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

## Collections

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

## 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

## Developing Views

- Creating Views
- Understanding View Engines
- Using the ASMX View Engine
- Using the Razor View Engines
- Using HTML Helpers
- Adding Validation
- Working with Strongly-Typed Views

## Routing Control

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

## Exception Handling

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

## 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

## Developing Controllers

- Creating Controllers
- Defining Action Methods
- Mapping URLs to Action Methods
- Understanding `ActionResult` Types
- Working with `ViewData` and `ViewBag`

## Developing Models

- Creating Model Classes
- Working with the Entity Framework
- Working with LINQ to SQL
- Using Scaffolding

## Integrating ASP.NET MVC and Web Forms

- What are Web Forms
- Using Web Forms in an MVC Application
- Using MVC in a Web Form Application

- Debugging Routes

- Linking to MVC Actions from Web Forms

## **MVC Unit Testing**

- Test-Driven Development
- Designing Test Cases
- Creating Unit Tests
- Using MS Test

## **Securing MVC Applications**

- ASP.NET Security
- Windows vs Forms Authentication
- Configuring Authentication
- Configuring Authorization
- Building a Secure Web Site
- Defending against Attacks
  - Cross-site Scripting
  - Session Hijacking
  - SQL Injection
  - Input Forgery

## **Deploying ASP.NET MVC Applications**

- Understanding Deployment Issues
- Required MVC Assemblies
- Server Requirements
- Configuring an ASP.NET Application for Deployment
- Using XCOPY Deployment
- Using WebDeploy

Hands On Technology Transfer  
The Best Way to Transfer Technology Skills

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

Copyright © 2021 Hands On Technology Transfer, Inc.