AV-001: Web Development with Visual Studio® 2012

Certificación MCSD® Web Applications

Detalles de la Carrera:

Duración:
140 horas.

Introducción:
Visual Studio 2012 facilita el ambiente para el desarrollo de aplicaciones convirtiéndose en tu mejor aliado, proporcionando muchas mejoras en la productividad e integrando herramientas que se adaptan a la creciente evolución tecnológica. Es por esta razón que encontrarás soporte para aplicaciones de Windows 8, aplicaciones Web y en la nube.

La capacitación en Visual Studio 2012 te permitirá desarrollar habilidades que te harán destacar como un desarrollador calificado experto en el uso de la herramienta, como actividad complementaria de la capacitación podrás participar en el proceso de certificación para tener garantía acerca del conocimiento obtenido.

Una certificación Microsoft en el mercado significa que eres un desarrollador con habilidades técnicas avanzadas y de alto valor para el mundo real, convirtiéndote en un colaborador ideal para lograr concretar las necesidades de un mercado en continua evolución

Certificación Relacionada:
Después de asistir a esta carrera, estarás listo para poder obtener la(s) siguientes certificación(es):

- **Microsoft Certified Solution Developer (MCSD®) Web Applications**

Contenido de la Carrera:

I. Module I: 20483 Programming in C#

About this Course:
This 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 Framework 4.5.

The course introduces many of the techniques and technologies employed by modern desktop and enterprise applications, including:

- Building new data types.
- Handling events.

- Programming the user interface.
- Accessing a database.
- Using remote data.
- Performing operations asynchronously.
- Integrating with unmanaged code.
- Creating custom attributes.
- Encrypting and decrypting data.

At the end of the course, students should leave the class with a solid knowledge of C# and how to use it to develop .NET Framework 4.5 applications.

This course uses Visual Studio 2012, running on Windows 8.

Audience Profile:

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.

At Course Completion:

After completing this course, students will be able to:

- Describe the core syntax and features of C#.
- Create and call methods, catch and 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 inheritance to create a class hierarchy, extend a .NET Framework class, and create generic classes and methods.
- 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 and update data.
- Use the types in the System.Net namespace and WCF Data Services to access and query remote data.
- 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
This module reviews the core syntax and features of the C# programming language. It also provides an introduction to the Visual Studio 2012 debugger.
Lessons
- Overview of Writing Applications using C#
- Datatypes, Operators, and Expressions
- C# Programming Language Constructs

Lab: Developing the Class Enrolment Application
- Implementing Edit Functionality for the Students List
- Implementing Insert Functionality for the Students List
- Implementing Delete Functionality for the Students List
- Displaying the Student Age

After completing this module, students will be able to:
- Describe the architecture of .NET Framework applications and use the features that Visual Studio 2012 and C# provide to support .NET Framework development.
- Use the basic data types, operators, and expressions provided by C#.
- Use standard C# programming constructs.

Module 2: Creating Methods, Handling Exceptions, and Monitoring Applications
This module explains how to create and call methods, catch and handle exceptions. This module also describes the monitoring requirements of large-scale applications.
Lessons
- Creating and Invoking Methods
- Creating Overloaded Methods and Using Optional and Output Parameters
- Handling Exceptions
- Monitoring Applications

Lab: Extending the Class Enrolment Application Functionality
- Refactoring the Enrolment Code
- Validating Student Information
- Saving Changes to the Class List
- After completing this module, students will be able to:
- Create and invoke methods, pass parameters to methods, and return values from methods.
- Create overloaded methods, and use optional parameters and output parameters.
- Catch and handle exceptions and write information to the event log.
- Explain the requirement for implementing logging, tracing, and profiling when building large-scale applications.

Module 3: Developing the Code for a Graphical Application
This module describes how to implement the basic structure and essential elements of a typical desktop application, including using structures and enumerations, collections, and events.
Lessons
- Implementing Structs and Enums

- Organizing Data into Collections
- Handling Events

Lab: Writing the Code for the Grades Prototype Application

- Adding Navigation Logic to the Application
- Creating Data Types to Store User and Grade Information
- Displaying User and Grade Information

After completing this module, students will be able to:

- Define and use structures and enumerations.
- Create and use simple collections for storing data in-memory.
- Create, subscribe to, and raise events.

Module 4: Creating Classes and Implementing Type-safe Collections

This module explains how to create classes, define and implement interfaces, and create and use generic collections. This module also describes the differences between value types and reference types in C#.

Lessons

- Creating Classes
- Defining and Implementing Interfaces
- Implementing Type-safe Collections

Lab: Adding Data Validation and Type-safety to the Grades Application

- Implementing the Teacher, Student, and Grade Types as Classes
- Adding Data Validation to the Grade Class
- Displaying Students in Name Order
- Enabling Teachers to Modify Class and Grade Data

After completing this module, students will be able to:

- Create and use custom classes.
- Define and implement custom interfaces.
- Use generics to implement type-safe collections.

Module 5: Creating a Class Hierarchy by Using Inheritance

This module explains how to use inheritance to create a class hierarchy and extend a .NET Framework class. This module also describes how to create generic classes and define extension methods.

Lessons

- Creating Class Hierarchies
- Extending .NET Framework Classes
- Creating Generic Types

Lab: Refactoring Common Functionality into the User Class

- Creating and Inheriting from the User Base Class
- Implementing Password Complexity by Using an Abstract Method
- Creating the ClassFullException Class

After completing this module, students will be able to:

- Define abstract classes and inherit from base classes to create a class hierarchy.
- Inherit from .NET Framework classes and use extension methods to add custom functionality to the inherited class.
- Create generic classes and methods.

Module 6: Reading and Writing Local Data
This module explains how to read and write data by using file input/output (I/O) and streams, and how to serialize and deserialize data in different formats.
Lessons
- Reading and Writing Files
- Serializing and Deserializing Data
- Performing I/O Using Streams

Lab: Generating the Grades Report
- Serializing the Data for the Grades Report as XML
- Previewing the Grades Report
- Persisting the Serialized Grades Data to a File

After completing this module, students will be able to:
- Read and write data to and from the file system by using file I/O.
- Convert data into a format that can be written to or read from a file or other data source.
- Use streams to send and receive data to or from a file or other data source.

Module 7: Accessing a Database
This module explains how to create and use an entity data model for accessing a database, and how to use LINQ to query and update data.
Lessons
- Creating and Using Entity Data Models
- Querying Data by Using LINQ
- Updating Data by Using LINQ

Lab: Retrieving and Modifying Grade Data
- Creating an Entity Model from the The School of Fine Arts Database
- Updating Student and Grade Data Using the Entity Framework
- Extending the Entity Model to Validate Data

After completing this module, students will be able to:
- Create an entity data model, describe the key classes contained in the model, and customize the generated code.
- Use LINQ to query and work with data.
- Use LINQ to insert, update, and delete data.

Module 8: Accessing Remote Data
This module explains how to use the types in the System.Net namespace, and WCF Data Services, to query and modify remote data.
Lessons
- Accessing Data Across the Web
- Accessing Data in the Cloud

Lab: Retrieving and Modifying Grade Data in the Cloud
- Creating a WCF Data Service for the SchoolGrades Database
- Integrating the WCF Data Service into the Application
- Retrieving Student Photographs Over the Web (if time permits)

After completing this module, students will be able to:

- Use the classes in the System.Net namespace to send and receive data across the Web.
- Create and use a WCF Data Service to access data in the cloud.

Module 9: Designing the User Interface for a Graphical Application
This module explains how to build and style a graphical user interface by using XAML. This module also describes how to display data in a user interface by using data binding.
Lessons
- Using XAML to Design a User Interface
- Binding Controls to Data
- Styling a User Interface

Lab: Customizing Student Photographs and Styling the Application
- Customizing the Appearance of Student Photographs
- Styling the Logon View
- Animating the StudentPhoto Control (If Time Permits)

After completing this module, students will be able to:
- Define XAML views and controls to design a simple graphical user interface.
- Use XAML data binding techniques to bind XAML elements to a data source and display data.
- Add styling and dynamic transformations to a XAML user interface.

Module 10: Improving Application Performance and Responsiveness
This module explains how to improve the throughput and response time of applications by using tasks and asynchronous operations.
Lessons
- Implementing Multitasking by using Tasks and Lambda Expressions
- Performing Operations Asynchronously
- Synchronizing Concurrent Access to Data

Lab: Improving the Responsiveness and Performance of the Application
- Ensuring that the User Interface Remains Responsive When Retrieving Data for Teachers
- Providing Visual Feedback During Long-Running Operations

After completing this module, students will be able to:
- Create tasks and lambda expressions to implement multitasking.
- Define and use asynchronous methods to improve application responsiveness.
- Coordinate concurrent access to data shared across multiple tasks by using synchronous primitives and concurrent collections.

Module 11: Integrating with Unmanaged Code
This module explains how to integrate unmanaged libraries and dynamic components into a C# application. This module also describes how to control the lifetime of unmanaged resources.
Lessons
- Creating and Using Dynamic Objects
- Managing the Lifetime of Objects and Controlling Unmanaged Resources

Lab: Upgrading the Grades Report
- Generating the Grades Report by Using Microsoft Office Word
- Controlling the Lifetime of Word Objects by Implementing the Dispose Pattern

After completing this module, students will be able to:

- Integrate unmanaged code into a C# application by using the Dynamic Language Runtime.
- Control the lifetime of unmanaged resources and ensure that they are disposed properly.

Module 12: Creating Reusable Types and Assemblies
This module explains how to examine the metadata of types by using reflection, create and use custom attributes, generate managed code at runtime, and manage different versions of assemblies.
Lessons
- Examining Object Metadata
- Creating and Using Custom Attributes
- Generating Managed Code
- Versioning, Signing and Deploying Assemblies

Lab: Specifying the Data to Include in the Grades Report
- Creating the IncludeInReport Attribute
- Generating the Report
- Storing the Grades.Utilities Assembly Centrally

After completing this module, students will be able to:
- Examine the metadata of objects at runtime by using reflection.
- Create and use custom attribute class.
- Generate managed code at runtime by using CodeDOM.
- Manage different versions of an assembly and deploy an assembly to the Global Assembly Cache.

Module 13: Encrypting and Decrypting Data
This module explains how to encrypt and decrypt data by using symmetric and asymmetric encryption.
Lessons
- Implementing Symmetric Encryption
- Implementing Asymmetric Encryption

Lab: Encrypting and Decrypting Grades Reports
- Encrypting the Grades Report
- Decrypting the Grades Report

After completing this module, students will be able to:
- Perform symmetric encryption by using the classes in the System.Security namespace.
- Perform asymmetric encryption by using the classes in the System.Security namespace

II. Module II: 20480 Programming in HTML5 with JavaScript and CSS3

About this Course:
This course provides an introduction to HTML5, CSS3, and JavaScript. This course helps students gain basic HTML5/CSS3/JavaScript programming skills. This course is an entry point into both the Web application and Windows Store apps training paths. The course focuses on using HTML5/CSS3/JavaScript to implement programming logic, define and use variables, perform looping and branching, develop user interfaces, capture and validate user input, store data, and create well-structured application.

The lab scenarios in this course are selected to support and demonstrate the structure of various application scenarios. They are intended to focus on the principles and coding components/structures that are used to establish an HTML5 software application.

This course uses Visual Studio 2012, running on Windows 8.

At Course Completion:

After completing this course, students will be able to:

- Explain how to use Visual Studio 2012 to create and run a Web application.
- Describe the new features of HTML5, and create and style HTML5 pages.
- Add interactivity to an HTML5 page by using JavaScript.
- Create HTML5 forms by using different input types, and validate user input by using HTML5 attributes and JavaScript code.
- Send and receive data to and from a remote data source by using XMLHTTPRequest objects and jQuery AJAX operations.
- Style HTML5 pages by using CSS3.
- Create well-structured and easily-maintainable JavaScript code.
- Use common HTML5 APIs in interactive Web applications.
- Create Web applications that support offline operations.
- Create HTML5 Web pages that can adapt to different devices and form factors.
- Add advanced graphics to an HTML5 page by using Canvas elements, and by using and Scalable Vector Graphics.
- Enhance the user experience by adding animations to an HTML5 page.
- Use Web Sockets to send and receive data between a Web application and a server.
- Improve the responsiveness of a Web application that performs long-running operations by using Web Worker processes.

Course Outline

Module 1: Overview of HTML and CSS

This module provides an overview of HTML and CSS, and describes how to use Visual Studio 2012 to build a Web application.

Lessons

- Overview of HTML
- Overview of CSS
- Creating a Web Application by Using Visual Studio 2012

Lab: Exploring the Contoso Conference Application

- Walkthrough of the Contoso Conference Application
- Examining and Modifying the Contoso Conference Application

After completing this module, students will be able to:

- Describe basic HTML elements and attributes.
- Explain the structure of CSS.
- Describe the tools available in Visual Studio 2012 for building Web applications.

Module 2: Creating and Styling HTML5 Pages

This module describes the new features of HTML5, and explains how to create and style HTML5 pages.

Lessons

- Creating an HTML5 Page
- Styling an HTML5 Page

Lab: Creating and Styling HTML5 Pages

- Creating HTML5 Pages
- Styling HTML5 Pages

After completing this module, students will be able to:

- Create static pages using the new features available in HTML5.
- Use CSS3 to apply basic styling to the elements in an HTML5 page.

Module 3: Introduction to JavaScript

This module provides an introduction to the JavaScript language, and shows how to use JavaScript to add interactivity to HTML5 pages.

Lessons

- Overview of JavaScript Syntax
- Programming the HTML DOM with JavaScript
- Introduction to jQuery

Lab: Displaying Data and Handling Events by Using JavaScript

- Displaying Data Programmatically
- Handling Events

After completing this module, students will be able to:

- Explain the syntax of JavaScript and describe how to use JavaScript with HTML5.
- Write JavaScript code that manipulates the HTML DOM and handles events.
- Describe how to use jQuery to simplify code that uses many common JavaScript APIs.

Module 4: Creating Forms to Collect Data and Validate User Input

This module describes the new input types available with HTML5, and explains how to create forms to collect and validate user input by using the new HTML5 attributes and JavaScript code.

Lessons

- Overview of Forms and Input Types
- Validating User Input by Using HTML5 Attributes
- Validating User Input by Using JavaScript

Lab: Creating a Form and Validating User Input

- Creating a Form and Validating User Input by Using HTML5 Attributes
- Validating User Input by Using JavaScript

After completing this module, students will be able to:

- Create forms that use the new HTML5 input types.
- Validate user input and provide feedback by using the new HTML5 attributes.
- Write JavaScript code to validate user input and provide feedback in cases where it is not suitable to use HTML5 attributes

Module 5: Communicating with a Remote Data Source

This module describes how to send and receive data to and from a remote data source by using an XMLHTTPRequest object and by performing jQuery AJAX operations.

Lessons

- Sending and Receiving Data by Using XMLHTTPRequest

- Sending and Receiving Data by Using jQuery AJAX operations

Lab: Communicating with a Remote Data Source

- Retrieving Data
- Serializing and Transmitting Data
- Refactoring the Code by Using jQuery ajax method

After completing this module, students will be able to:

- Serialize, deserialize, send, and receive data by using XMLHTTPRequest objects.
- Simplify code that serializes, deserializes, sends, and receives data by using the jQuery ajax method

Module 6: Styling HTML5 by Using CSS3

This module describes how to style HTML5 pages and elements by using the new features available in CSS3.

Lessons

- Styling Text
- Styling Block Elements
- CSS3 Selectors
- Enhancing Graphical Effects by Using CSS3

Lab: Styling Text and Block Elements using CSS3

- Styling the Navigation Bar
- Styling the Page Header
- Styling the About Page

After completing this module, students will be able to:

- Style text elements on an HTML5 page by using CSS3.
- Apply styling to block elements by using CSS3.
- Use CSS3 selectors to specify the elements to be styled in a Web application.
- Implement graphical effects and transformations by using the new CSS3 properties.

Module 7: Creating Objects and Methods by Using JavaScript

This module explains how to write well-structured and easily-maintainable JavaScript code, and how to apply object-oriented principles to JavaScript code in a Web application.

Lessons

- Writing Well-Structured JavaScript
- Creating Custom Objects
- Extending Objects

Lab: Refining Code for Maintainability and Extensibility

- Inheriting From Objects
- Refactoring Code to Use Objects

After completing this module, students will be able to:

- Describe the benefits of structuring JavaScript code carefully to aid maintainability and extensibility.
- Explain best practices for creating custom objects in JavaScript.
- Describe how to extend custom and native objects to add functionality.

Module 8: Creating Interactive Pages using HTML5 APIs

This module describes how to use some common HTML5 APIs to add interactive features to a Web application. This module also explains how to debug and profile a Web application.

Lessons

- Interacting with Files
- Incorporating Multimedia
- Reacting to Browser Location and Context
- Debugging and Profiling a Web Application

Lab: Creating Interactive Pages by Using HTML5 APIs

- Incorporating Video
- Incorporating Images
- Using the Geolocation API

After completing this module, students will be able to:

- Use the Drag and Drop, and the File APIs to interact with files in a Web application.
- Incorporate audio and video into a Web application.
- Detect the location of the user running a Web application by using the Geolocation API.
- Explain how to debug and profile a Web application by using the Web Timing API and the Internet Explorer Developer Tools.

Module 9: Adding Offline Support to Web Applications

This module describes how to add offline support to a Web application, to enable the application to continue functioning in a user's browser even if the browser is disconnected from the network.

Lessons

- Reading and Writing Data Locally
- Adding Offline Support by Using the Application Cache

Lab: Adding Offline Support to a Web Application

- Implementing the Application Cache
- Implementing Local Storage

After completing this module, students will be able to:

- Save and retrieve data locally on the user's computer by using the Local Storage API.
- Provide offline support for a Web application by using the Application Cache API.

Module 10: Implementing an Adaptive User Interface

This module describes how to create HTML5 pages that can dynamically detect and adapt to different devices and form factors.

Lessons

- Supporting Multiple Form Factors
- Creating an Adaptive User Interface

Lab: Implementing an Adaptive User Interface

- Creating a Print-Friendly Stylesheet
- Adapting Page Layout To Fit a Different Form Factor

After completing this module, students will be able to:

- Describe the need to detect device capabilities and react to different form factors in a Web application.
- Create a Web page that can dynamically adapt its layout to match different form factors.

Module 11: Creating Advanced Graphics
This module describes how to create advanced graphics for an HTML5 Web application by using a Canvas element, and by using Scalable Vector Graphics.
Lessons
- Creating Interactive Graphics by Using Scalable Vector Graphics
- Programmatically Drawing Graphics by Using a Canvas

Lab: Creating Advanced Graphics
- Creating an Interactive Venue Map by Using Scalable Vector Graphics
- Creating a Speaker Badge by Using a Canvas Element

After completing this module, students will be able to:
- Use Scalable Vector Graphics to add interactive graphics to an application.
- Draw complex graphics on an HTML5 Canvas element by using JavaScript code.

Module 12: Animating the User Interface
This module describes how to enhance the user experience in an HTML5 Web application by adding animations.
Lessons
- Applying CSS Transitions
- Transforming Elements
- Applying CSS Key-frame Animations

Lab: Animating User Interface Elements
- Applying Transitions to User Interface Elements
- Applying Key-Frame Animations

After completing this module, students will be able to:
- Apply CSS transitions to elements on an HTML5 page, and write JavaScript code to detect when a transition has occurred.
- Describe the different types of 2D and 3D transitions available with CSS3
- Implement complex animations by using CSS key-frames and JavaScript code.

Module 13: Implementing Real-Time Communications by Using Web Sockets
This module explains how to use Web Sockets to transmit and receive data between an HTML5 Web application and a server.
Lessons
- Introduction to Web Sockets
- Sending and Receiving Data by Using Web Sockets

Lab: Implementing Real-Time Communications by Using Web Sockets
- Receiving Data from Web Socket
- Sending Data to a Web Socket
- Sending Multiple Types of Messages To or From a Web Socket

After completing this module, students will be able to:
- Explain how Web Sockets work and describe how to send and receive data through a Web Socket.
- Use the Web Socket API with JavaScript to connect to a Web Socket server, send and receive data, and handle the different events that can occur when a message is sent or received.

Module 14: Creating a Web Worker Process

This module describes how to use Web Worker Processes to perform long-running operations asynchronously and improve the responsiveness of an HTML5 Web application.

Lessons

- Introduction to Web Workers
- Performing Asynchronous Processing by Using a Web Worker

Lab: Creating a Web Worker Process

- Improving Responsiveness by Using a Web Worker

After completing this module, students will be able to:

- Describe the purpose of a Web Worker process, and how it can be used to perform asynchronous processing as well as provide isolation for sensitive operations.
- Use the Web Worker APIs from JavaScript code to create, run, and monitor a Web Worker process.

III. Module III: 20486 Developing ASP.NET MVC 4 Web Applications

About this course

In this course, students will learn to develop advanced ASP.NET MVC applications using .NET Framework 4.5 tools and technologies. The focus will be on coding activities that enhance the performance and scalability of the Web site application. ASP.NET MVC will be introduced and compared with Web Forms so that students know when each should/could be used. This course will also prepare the student for exam 70-486.

Audience Profile

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.

At Course Completion

After completing this course, students will be able 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.
- Create MVC Models and write code that implements business logic within Model methods, properties, and events.
- Add Controllers to an MVC Application to manage user interaction, update models, and select and return Views.
- Create Views in an MVC application that display and edit data and interact with Models and Controllers.
- Run unit tests and debugging tools against a web application in Visual Studio 2012 and configure an application for troubleshooting.

- Develop a web application that uses the ASP.NET routing engine to present friendly URLs and a logical navigation hierarchy to users.
- Implement a consistent look and feel, including corporate branding, across an entire MVC web application.
- Use partial page updates and caching to reduce the network bandwidth used by an application and accelerate responses to user requests.
- Write JavaScript code that runs on the client-side and utilizes the jQuery script library to optimize the responsiveness of an MVC web application.
- Implement a complete membership system in an MVC 4 web application.
- Build an MVC application that resists malicious attacks and persists information about users and preferences.
- Describe how to write a Windows Azure web service and call it from and MVC application.
- Describe what a Web API is and why developers might add a Web API to an application.
- Modify the way browser requests are handled by an MVC application.
- Describe how to package and deploy an ASP.NET MVC 4 web application from a development computer to a web server for staging or production.

Course Outline

Module 1: Exploring ASP.NET MVC 4
The goal of this module is to outline to the students the components of the Microsoft Web Technologies stack, which can be used to host a completed web application. Students will also learn about ASP.NET 4.5 and be introduced to the web forms, web pages, and MVC programming models. Finally they will see an overview of ASP.NET MVC 4, including new features and configuration.
Lessons
- Overview of Microsoft Web Technologies
- Overview of ASP.NET 4.5
- Introduction to ASP.NET MVC 4

Lab: Exploring ASP.NET MVC 4
- Exploring a Photo Sharing Application
- Exploring a Web Pages Application
- Exploring a Web Forms Application
- Exploring an MVC Application

After completing this module, students will be able to:
- Describe the Microsoft Web Technologies stack and select an appropriate technology to use to develop any given application.

Module 2: Designing ASP.NET MVC 4 Web Applications
The goal of this module is to introduce students to the typical design process that architects must complete when they plan an MVC 4 application. At this stage in the design process, MVC 4 has been selected as the most appropriate programming model, but the details of the application, such as the overall architecture, Controllers, Views, Models, and routes to create, have not been fixed. How to plan such details is shown during this module.
Lessons
- Planning in the Project Design Phase

- Designing Models, Controllers, and Views

Lab: Designing ASP.NET MVC 4 Web Applications

- Planning Models
- Planning Controllers
- Planning Views
- Architecting an MVC Application

After completing this module, students will be able to:

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

Module 3: Developing ASP.NET MVC 4 Models

The goal of this module is to enable the students to create Models within an MVC application that implement the business logic necessary to satisfy business requirements. The module also describes how to implement a connection to a database, or alternative data store, using the Entity Framework and LINQ.

Lessons

- Creating MVC Models
- Working with Data

Lab: Developing ASP.NET MVC 4 Models

- Creating an MVC Project and Adding a Model
- Creating a New SQL Azure Database in Visual Studio
- Adding Properties and Methods to MVC Models
- Using Display and Edit Annotations in MVC Models

After completing this module, students will be able to:

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

Module 4: Developing ASP.NET MVC 4 Controllers

The goal of this module is to enable students to add Controllers to MVC applications and to implement actions that respond to user input and other events. The students will learn how Controllers relate to Models and how to implement Controller actions that define the View used to display or edit data. This module also covers how to write action filters that run code before or after multiple actions in the Controller. The students will learn about situations when action filters are useful.

Lessons

- Writing Controllers and Actions
- Writing Action Filters

Lab: Developing ASP.NET MVC 4 Controllers

- Adding an MVC Controller and Writing the Actions
- Writing the Action Filters in a Controller
- Using the Photo Controller

After completing this module, students will be able to:

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

Module 5: Developing ASP.NET MVC 4 Views

The goal of this module is to describe the role of Views in an MVC web application and enable users to create and code them. The syntax of a Razor View is of critical importance for students to understand because it defines both the layout and the functionality of the data display. HTML Helpers will also be discussed in detail and common Helpers, such as Html.ActionLink() and Html.EditorFor(), will be described. Reusing code by defining Partial Views and Razor Helpers will be discussed as well.

Lessons

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

Lab: Developing ASP.NET MVC 4 Views

- Adding a View for Photo Display
- Adding a View for New Photos
- Creating and Using a Partial View
- Adding a Home View and Testing the Views

After completing this module, students will be able to:

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

Module 6: Testing and Debugging ASP.NET MVC 4 Web Applications

The goal of this module is to enable students to increase the resilience and quality of an application by locating and correcting code errors, bugs, and other unexpected results. MVC applications are well suited to unit testing techniques and these techniques ensure a high quality of code by systematically testing the functionality of each small component. In addition the debugging tools and exception handling available in Visual Studio will be explained.

Lessons

- Unit Testing MVC Components
- Implementing an Exception Handling Strategy

Lab: Testing and Debugging the ASP.NET MVC 4 Web Applications

- Performing Unit Tests
- Configuring Exception Handling

After completing this module, students will be able to:

- Run unit tests and debugging tools against a web application in Visual Studio 2012 and configure an application for troubleshooting.

Module 7: Structuring ASP.NET MVC 4 Web Applications

The goal of this module is to enable students to structure a web application in such a way that users can rapidly locate the information they need. Two aspects of the design are emphasized: the URLs presented in the browser address bar should be understandable and can be controlled by adding routes to the ASP.NET Routing Engine, and the navigation controls, such as menus and breadcrumb trails, should present the most relevant links to frequently read pages. Search Engine Optimization is important throughout this module.

Lessons

- Analyzing Information Architecture
- Configuring Routes
- Creating a Navigation Structure

Lab: Structuring ASP.NET MVC 4 Web Applications
- Using the Routing Engine
- Building Navigation Controls

After completing this module, students will be able to:
- Develop a web application that uses the ASP.NET routing engine to present friendly URLs and a logical navigation hierarchy to users.

Module 8: Applying Styles to ASP.NET MVC 4 Web Applications
The goal of this module is to explore how students can impose a consistent look and feel to an MVC application and share other common components, such as headers and footers, between all Views. Besides describing CSS styles and template views, the module will discuss how to migrate a look and feel created by a web designer into an MVC application. Techniques for adapting the display of a site for small screens and mobile devices will also be introduced.
Lessons
- Using Template Views
- Applying CSS to an MVC Application
- Creating an Adaptive User Interface

Lab: Applying Styles to ASP.NET MVC 4 Web Applications
- Using Template Views
- Applying a Consistent Look and Feel to an MVC Application
- Adapting Webpages for Different Browsers

After completing this module, students will be able to:
- Implement a consistent look and feel, including corporate branding, across an entire MVC web application.

Module 9: Building Responsive Pages in ASP.NET MVC 4 Web Applications
The goal of this module is to describe to the students how partial page updates and caching can optimize the responsiveness of a web application. Students will see how to make use of AJAX helpers and partial views to update small portions of a page instead of refreshing the entire page. The module also covers the different caches developers can use to store rendered pages and discusses how to configure caching for maximum performance.
Lessons
- Using AJAX and Partial Page Updates
- Implementing a Caching Strategy

Lab: Building Responsive Pages in ASP.NET MVC 4 Web Applications
- Using Partial Page Updates
- Configuring ASP.NET Caches

After completing this module, students will be able to:
- Use partial page updates and caching to reduce the network bandwidth used by an application and accelerate responses to user requests.

Module 10: Using JavaScript and jQuery for Responsive MVC 4 Web Applications
The goal of this module is to teach the students techniques that run code on the browser. This approach can increase the responsiveness of the application because a rendered page can respond to a user action

without reloading the entire page from the server. Students will learn about the jQuery script library and how to use it to call web services and update user interface components.

Lessons

- Rendering and Running JavaScript Code
- Using jQuery and jQueryUI

Lab: Using JavaScript and jQuery for Responsive MVC 4 Web Applications

- Using jQuery to Respond to Users
- Using jQueryUI to Build a User Interface

After completing this module, students will be able to:

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

Module 11: Controlling Access to ASP.NET MVC 4 Web Applications

The goal of this module is to ensure good security in terms of strong authentication and authorization for access. The lessons describe how to enable anonymous users to create their own user account and gain privileged access to content.

Lessons

- Implementing Authentication and Authorization
- Assigning Roles and Membership

Lab: Controlling Access to ASP.NET MVC 4 Web Applications

- Configuring Authentication
- Controlling Access to Resources
- Providing User Account Facilities

After completing this module, students will be able to:

- Implement a complete membership system in an MVC 4 web application.

Module 12: Building a Resilient ASP.NET MVC 4 Web Application

The goal of this module is to enable the students to build applications that are stable and reliable. Such applications are not vulnerable to common hacking techniques such as cross-site scripting and also store state information such as the contents of a shopping cart and user preferences. This state information is preserved when servers or browsers restart, connections are lost, and other connectivity issues occur.

Lessons

- Developing Secure Sites
- State Management

Lab: Building a Resilient ASP.NET MVC 4 Web Application

- Storing User Preferences
- Using User Preferences in the Photo Gallery

After completing this module, students will be able to:

- Build an MVC application that resists malicious attacks and persists information about users and preferences.

Module 13: Using Windows Azure Web Services in ASP.NET MVC 4 Web Applications

The goal of this module is to introduce Windows Azure to the students and explain why a developer would write a Windows Azure service instead of code in a web application. Students will also see how to write such a service and call it from a web application or from other applications, such as a mobile device app.

Lessons

- Introduction to Windows Azure
- Designing and Writing Windows Azure Services
- Consuming Windows Azure Services in a Web Application

Lab: Using Windows Azure Web Services in ASP.NET MVC 4 Web Applications

- Creating and Coding a Windows Azure Service
- Consuming Data from a Windows Azure Service

After completing this module, students will be able to:

- Describe how to write a Windows Azure web service and call it from and MVC application.

Module 14: Implementing Web APIs in ASP.NET MVC 4 Web Applications

The goal of the module is to introduce the concept of a Web API to students and to describe how to make an application's core functionality more broadly available for integration into other web and mobile applications. Students will learn about the new Web API feature of MVC 4 and see how to build a RESTful Web API and call it from other applications.

Lessons

- Developing a Web API
- Calling a Web API from Mobile and Web Applications

Lab: Implementing Web APIs in ASP.NET MVC 4 Web Applications

- Developing a Web API in MVC 4
- Adding Routes and Controllers to Handle REST Requests
- Calling RESTful services from Client-Side Code

After completing this module, students will be able to:

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

Module 15: Handling Requests in ASP.NET MVC 4 Web Applications

The goal of this module is to describe how to write components that intercept requests from browsers before they are received by MVC Controllers. These components include HTTP Modules, HTTP Handlers, and the Web Sockets protocol. The module describes scenarios in which developers use such components and shows how to add them to an MVC application.

Lessons

- Using HTTP Modules and HTTP Handlers
- Using Web Sockets

Lab: Handling Requests in ASP.NET MVC 4 Web Applications

- Writing a Web Handler that Uses Web Sockets
- Building a Chat Room in the Photo Sharing Application

After completing this module, students will be able to:

- Modify the way browser requests are handled by an MVC application.

Module 16: Deploying ASP.NET MVC 4 Web Applications

The goal for this module is to enable students to deploy a completed MVC application to a web server or Windows Azure. The module begins by describing testing, staging, and production deployments and the web server environments required for each. It also describes the advantages and disadvantages of using Windows Azure to host the application. Students also see all the available deployment options in Visual Studio.

Lessons

- Deploying Web Applications
- Deploying MVC 4 Applications

Lab: Deploying ASP.NET MVC 4 Web Applications

- Deploying an Application to Windows Azure
- Testing the Completed Application

After completing this module, students will be able to:

- Describe how to package and deploy an ASP.NET MVC 4 web application from a development computer to a web server for staging or production

IV. Module IV: 20487 Developing Windows Azure and Web Services

About this Course

In this course, students will learn how to design and develop services that access local and remote data from various data sources. Students will also learn how to develop and deploy services to hybrid environments, including on-premises servers and Windows Azure. This course helps people prepare for exam 70-487.

In this course, students will learn how to design and develop services that access local and remote data from various data sources. Students will also learn how to develop and deploy services to hybrid environments, including on-premises servers and Windows Azure. This course helps people prepare for exam 70-487.

Audience Profile

This course is intended for both novice and experienced .NET developers who have a minimum of six months programming experience, and want to learn how to develop services and deploy them to hybrid environments.

At Course Completion

After completing this course, students will be able to:

- Query and manipulate data with Entity Framework
- Use ASP.NET Web API to create HTTP-based services and consume them from .NET and non-.NET clients
- Extend ASP.NET Web API services using message handlers, model binders, action filters, and media type formatters
- Create SOAP-based services with the Windows Communication Foundation (WCF) and consume them from .NET clients
- Apply design principles to service contracts and extend WCF services using custom runtime components and behaviors
- Secure WCF services using transport and message security
- Use Windows Azure Service Bus for relayed messaging and brokered messaging using queues and topics
- Host services on on-premises servers, and on various Windows Azure environments, such as Web Roles, Worker Roles, and Web Sites
- Deploy services to both on-premises servers and Windows Azure

- Store and access data in Windows Azure Storage, and configure storage access rights
- Monitor and log services, both on-premises and in Windows Azure
- Implement federated authentication by using ACS with ASP.NET Web API services
- Create scalable, load-balanced services

Course Outline

Module 1: Overview of service and cloud technologies

This module describes the Microsoft data, service, and cloud stacks. It also describes the various components that comprise Windows Azure.

Lessons

- Key Components of Distributed Applications
- Data and Data Access Technologies
- Service Technologies
- Cloud Computing
- Exploring Blue Yonder Airlines' Travel Companion Application

Lab: Exploring the work environment

- Create a Windows Azure SQL Database
- Create an Entity Data Model
- Create an ASP.NET Web API service
- Deploy a web application to Windows Azure

After completing this module, students will be able to:

- Describe the overall architecture of distributed applications.
- Describe the data platform technologies supported by Microsoft.
- Describe the different approaches and technologies used for developing services.
- Describe cloud computing concepts and the Windows Azure ecosystem.

Module 2: Querying and manipulating data using Entity Framework

This module explains how to create Entity Framework models and use them to query and manipulate data.

Lessons

- ADO.NET overview
- Creating an entity data model
- Querying data
- Manipulating data

Lab: Creating a data access layer using Entity Framework

- Explore the data model and integration test projects
- Create a data model
- Query and manipulate data

After completing this module, students will be able to:

- Describe how to use ADO.NET to query and manipulate data.
- Create entity data models using the different design approaches of Entity Framework.
- Query a database using various Entity Framework techniques.
- Manipulate data by using Entity Framework.

Module 3: Creating and consuming ASP.NET Web API services
This module explains how to create HTTP based services using the ASP.NET Web API.
Lessons
- What are HTTP services?
- Creating an ASP.NET Web API service
- Handling HTTP requests and responses
- Hosting and consuming ASP.NET Web API services

Lab: Creating the travel reservation ASP.NET Web API service
- Create an ASP.NET Web API service
- Consume an ASP.NET Web API service

After completing this module, students will be able to:
- Describe the HTTP protocol and how it is used with REST.
- Create a basic ASP.NET Web API service by using routing, controllers, and actions.
- Convert HTTP request content to .NET objects and convert return values to responses.
- Host and consume ASP.NET Web API services in various server and client scenarios.

Module 4: Extending and securing ASP.NET Web API services
This module explains how to extend and secure ASP.NET web API services to support real world scenarios.
Lessons
- The ASP.NET Web API request pipeline
- The ASP.NET Web API response pipeline
- Creating OData services
- Implementing Security in ASP.NET Web API services
- Injecting dependencies into controllers

Lab: Extending Travel Companion's ASP.NET Web API services
- Create a dependency resolver for repositories
- Add a new media type for RSS requests
- Add OData capabilities to the flight schedule service
- Apply validation rules in the booking service
- Secure the communication between client and server

After completing this module, students will be able to:
- Describe how messages flow through the ASP.NET Web API request processing pipeline.
- Describe how messages flow through the ASP.NET Web API response processing pipeline.
- Create ASP.NET Web API OData services.
- Implement security in ASP.NET Web API services.
- Create a dependency resolver that injects dependencies into ASP.NET Web API controllers.

Module 5: Creating WCF services
This module explains how to create WCF services, host them, and consume them from other applications.
Lessons
- Advantages of creating services with WCF
- Creating and implementing a contract
- Configuring and hosting WCF services

- Consuming WCF services

Lab: Creating and consuming the WCF booking service

- Create the WCF booking service
- Configure and host the WCF service
- Consume the WCF service from the ASP.NET Web API booking service

After completing this module, students will be able to:

- Describe why and when to use WCF to create services.
- Implement a service using contracts.
- Host a WCF service with endpoint configuration in code and configuration file.
- Consume a WCF services from .NET clients.

Module 6: Designing and extending WCF services

This module explains how to design a WCF service contracts with duplex support, async operations, and one-way operations. It also explains how to create services that use various instancing and concurrency modes. In addition, it describes how to extend a WCF service with custom behaviors and runtime components.

Lessons

- Applying design principles to service contracts
- Handling distributed transactions
- WCF pipeline architecture
- Extending the WCF pipeline

Lab: Designing and extending WCF services

- Create a custom error handler runtime component
- Add support for distributed transactions to the WCF booking service
- Use asynchronous WCF client calls

After completing this module, students will be able to:

- Create service contracts that support service design principles.
- Create services that support distributed transactions.
- Describe the architecture of the WCF pipeline and how to control it with behaviors.
- Extend WCF with runtime components and extensible objects.

Module 7: Implementing Security in WCF services

This module explains how to implement security in WCF services by using transport and message security. It also describes how to configure and implement authentication and authorization for a service

Lessons

- Transport security
- Message security
- Configuring service authentication and authorization

Lab: Securing a WCF service

- Secure the WCF service
- Configure the ASP.NET Web API booking service for secured communication

After completing this module, students will be able to:

- Configure a service for transport security.
- Configure a service for message security.

- Authenticate and authorize users.

Module 8: Windows Azure Service Bus
This module explains how to use the Windows Azure Service Bus for advanced routing and messaging scenarios.
Lessons
- Windows Azure Service Bus Relays
- Windows Azure Service Bus Queues
- Windows Azure Service Bus Topics

Lab: Windows Azure Service Bus
- Use a service bus relay for the WCF booking service
- Publish booking updates to clients using Windows Azure Service Bus Topics

After completing this module, students will be able to:
- Connect hybrid environments with Windows Azure Service Bus Relays.
- Use brokered messaging with Windows Azure Service Bus queues.
- Use subscription-based messaging with Windows Azure Service Bus topics.

Module 9: Hosting services
This module explains how to host services on various Windows Azure environments, such as Web Roles, Worker Roles, and Web Sites
Lessons
- Hosting services on-premises
- Hosting services in Windows Azure

Lab: Hosting Services
- Host the WCF booking service in IIS
- Host the ASP.NET Web API services in a Windows Azure Web role
- Host the booking management service in a Windows Azure Web Site

After completing this module, students will be able to:
- Describe the common on-premises hosting environments.
- Host a service in Windows Azure hosting environments.

Module 10: Deploying Services
This module explains how to deploy services to both on-premises and cloud environments.
Lessons
- Web Deployment with Visual Studio
- Creating and deploying Web Application packages
- Command-line tools for web deployment packages
- Deploying to Windows Azure
- Continuous delivery with TFS and GIT
- Best practices for production deployment

Lab: Deploying services
- Deploying an updated service to Windows Azure
- Updating a Windows Azure Web Site with Web Deploy
- Exporting and importing an IIS deployment package

After completing this module, students will be able to:

- Deploy services from Visual Studio.
- Deploy services by using web deployment packages.
- Deploy services using command-line tools.
- Deploy services to Windows Azure environments.
- Ensure that Windows Azure deployments are up-to-date with continuous delivery.

Module 11: Windows Azure Storage
This module explains how to store and access data stored in Windows Azure Storage. It also explains how to configure storage access rights for storage containers and content.
Lessons
- Introduction to Windows Azure storage
- Windows Azure Blob Storage
- Windows Azure Table Storage
- Windows Azure Queue Storage
- Restricting access to Windows Azure Storage

Lab: Windows Azure Storage
- Storing content in Windows Azure storage
- Accessing Windows Azure storage
- Creating shared access signatures for blobs

After completing this module, students will be able to:
- Describe the reasons for using Windows Azure storage.
- Use blobs for storing resources.
- Use tables for storing structured, non-relational data.
- Use queues for sending and receiving messages asynchronously.
- Configure access level and shared access signatures for Windows Azure Storage services.

Module 12: Monitoring and diagnostics
This module explains how to monitor and log services, both on-premises and in Windows Azure
Lessons
- Performing diagnostics using tracing
- Configuring service diagnostics
- Monitoring IIS
- Monitoring services using Windows Azure diagnostics
- Debugging using IntelliTrace
- Collecting Windows Azure metrics

Lab: Monitoring and Diagnostics
- Configuring WCF tracing and message logging
- Configuring Windows Azure diagnostics

After completing this module, students will be able to:
- Write diagnostics trace messages.
- Configure and monitor service diagnostic information.
- Monitor IIS-hosted services.
- Monitor Windows Azure applications using Windows Azure diagnostics.
- Debug services with IntelliTrace.

- Collect Windows Azure metrics.

Module 13: Identity management and access control
This module describes claim-based identity concepts and standards, and how to implement federated authentication by using ACS to secure an ASP.NET Web API service. It also explains how to use ACS to secure Windows Azure Service Bus connections.
Lessons
- Claim-based identity concepts
- Access Control Service
- Configuring services to use federated identities
- Handling federated identities in the client side

Lab: Identity management and access control
- Configure Windows Azure ACS
- Integrate ACS with the ASP.NET Web API
- Examine the authentication procedure in the client application

After completing this module, students will be able to:
- Describe claim-based identity concepts.
- Describe the Access Control Service and its purpose.
- Configure a service to require federated identities.
- Configure a service client with federated identity

Module 14: Scaling Services
This module explains how to create scalable services and applications.
Lessons
- Introduction to scalability
- Load balancing
- Scaling on-premises services with distributed cache
- Windows Azure caching
- Caveats of scaling services
- Scaling globally

Lab: Scalability
- Use Windows Azure Caching
- Support federated security in a scaled environment

After completing this module, students will be able to:
- Describe the reasons and techniques for scaling services.
- Describe how load balancing can be used with on-premises and Windows Azure environments.
- Integrate a distributed cache mechanism into a service by using Windows Server AppFabric Cache.
- Describe the distributed cache solutions offered by Windows Azure.
- Understand the caveats of scaling out services and how to resolve them.
- Scale Windows Azure solutions outside of the data center