

Microsoft Web Apps Developer Boot Camp Courses

**Bootcamp Title – MCSD: Web Apps Developer (1 Cert)**   
Number of Days – 10  
Number of Exams – 3  
Number of Certifications – 1  
Cost - $6,795.00

Certifications:

MCSD: Web Apps

Exams:

**70-480:** Programming in HTML5 with JavaScript and CSS3

**70-486:** Designing ASP .Net 4.5 MVC Web Applications

**70-487:** Developing Windows Azure and Web Services

Course Description:

The MCSD Web Applications certification boot camp is a 9 day comprehensive deep dive into Web Apps covering topics such as writing programming, developing and creating. This instructor led face to face training camp will teach you the skills needed to support a Web Apps environment.

Class Objectives (*Following information customized from Microsoft Learning Test Objectives)*

**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**

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**

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**

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**

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**

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**

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**

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**

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**

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**

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**

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**

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**

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**

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.

**Module 1: Exploring ASP.NET MVC4**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 MVC4**

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**

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**

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**

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**

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 ASP.NET MVC 4 Web Applications**

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**

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**

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**

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 Executing JavaScript Code
* Using jQuery and jQueryUI

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

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 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**

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**

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**

* Introducing 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**

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**

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**

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 a Web Application
* Deploying an MVC 4 Application

**Lab : Deploying ASP.NET MVC 4 Web Applications**

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.

|  |
| --- |
| **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**  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**  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**  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**  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**  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**  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**  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**  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**  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**  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**  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**  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**  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**  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. |
|  |