70-487:
Developing Windows Azure and Web Services

The following tables show where changes to exam 70-487 have been made to include updates that relate to Windows Azure and Visual Studio 2013 tasks. These changes are effective as of April 30, 2014.

1. Accessing data

Tasks currently measured	Tasks Added/Changed post *April 2014*
Choose data access technologies Choose a technology (ADO.NET, Entity Framework, WCF Data Services) based on application requirements	Modified subtask: Choose a technology (ADO.NET, Entity Framework, WCF Data Services, Azure storage) based on application requirements
Implement caching Cache static data, apply cache policy (including expirations); use CacheDependency to refresh cache data; query notifications	No Change
Implement transactions Manage transactions by using the API from System.Transactions namespace; implement distributed transactions; specify transaction isolation level	No Change
Implement data storage in Windows Azure Access data storage in Windows Azure; choose data storage mechanism in Windows Azure (blobs, tables, queues, SQL Database); distribute data by using the Content delivery network (CDN); handle exceptions by using retries (SQL Database); manage Windows Azure Caching	No Change
Create and implement a WCF Data Services service Address resources; implement filtering; create a query expression; access payload formats (including JSON); use data service interceptors and service operators	No Change
Manipulate XML data structures	No Change

Read filter, create, modify XML data structures; Manipulate XML data by using XMLReader, XMLWriter, XMLDocument, XPath, LINQ to XML; transform XML by using XSLT transformations	

2. Querying and manipulating data by using the Entity Framework

Tasks currently measured	Tasks Added/Changed post *April 2014*
Query and manipulate data by using the Entity Framework Query, update, and delete data by using DbContext; build a query that uses deferred execution; implement lazy loading and eager loading; create and run compiled queries; query data by using Entity SQL	Added subtasks: Perform asynchronous operations using Entity Framework; map a stored procedure
Query and manipulate data by using Data Provider for Entity Framework Query and manipulate data by using Connection, DataReader, Command from the System.Data.EntityClient namespace; perform synchronous and asynchronous operations; manage transactions (API)	Added subtask: Programmatically configure a Data Provider
Query data by using LINQ to Entities Query data by using LINQ operators (for example, project, skip, aggregate, filter, and join); log queries; implement query boundaries (IQueryable vs. IEnumerable)	Modified subtask: Log queries and database commands Added subtask: Implement async query
Query and manipulate data by using ADO.NET Query and manipulate data by using Connection, DataReader, Command, DataAdapter, DataSet; perform synchronous and asynchronous operations; manage transactions (API)	No Change
Create an Entity Framework data model Structure the data model using table per type, table per class, table per hierarchy; choose and implement an approach to manage a data model (code first vs. model first vs. database first); implement POCO objects; describe a data model by using conceptual schema definitions, storage schema definition, and mapping language (CSDL, SSDL, MSL)	Modified subtask: Describe a data model by using conceptual schema definitions, storage schema definition, mapping language (CSDL, SSDL, MSL), and Custom Code First Conventions

3. Designing and implementing WCF Services

Tasks currently measured	Tasks Added/Changed post *April 2014*
Create a WCF service Create contracts (service, data, message, callback, and fault); implement message inspectors; implement asynchronous operations in the service	No Change
Configure WCF services by using configuration settings Configure service behaviors; configure service endpoints; configure binding; specify a service contract; expose service metadata (XSDs, WSDL, and metadata exchange endpoint)	Modified subtask: Configure bindings (including WebSocket bindings) Added subtask: Configure message compressions and encoding
Configure WCF services by using the API Configure service behaviors; configure service endpoints; configure binding; specify a service contract; expose service metadata (XSDs, WSDL, and metadata exchange); WCF routing and discovery features	No Change
Secure a WCF service Implement message level security, implement transport level security; implement certificates	Added subtask: Design and implement multiple authentication modes
Consume WCF services Generate proxies by using SvcUtil; generate proxies by creating a service reference; create and implement channel factories	No Change
Version a WCF service Version different types of contracts (message, service, data); configure address, binding, and routing service versioning	No Change
Create and configure a WCF service on Windows Azure Create and configure bindings for WCF services (Azure SDK—extensions to WCF); relay bindings to Azure using service bus endpoints; integrate with the Azure service bus relay	No Change
Implement messaging patterns Implement one way, request/reply, streaming, and duplex communication; implement Windows Azure Service Bus and Windows Azure Queues	No Change
Host and manage services Manage services concurrency (single, multiple, reentrant); create service hosts; choose a hosting mechanism; choose an instancing mode (per call, per session, singleton); activate and manage a service by using AppFabric; implement	No Change

transactional services; host services in a Windows Azure worker role	

4. Creating and consuming Web API-based services

Tasks currently measured	Tasks Added/Changed post *April 2014*
Design a Web API Define HTTP resources with HTTP actions; plan appropriate URI space, and map URI space using routing; choose appropriate HTTP method (get, put, post, delete) to meet requirements; choose appropriate format (Web API formats) for responses to meet requirements; plan when to make HTTP actions asynchronous	Added subtask: Design and implement routes
Implement a Web API Accept data in JSON format (in JavaScript, in an AJAX callback); use content negotiation to deliver different data formats to clients; define actions and parameters to handle data binding; use HttpMessageHandler to process client requests and server responses; implement dependency injection, along with the dependency resolver, to create more flexible applications; implement action filters and exception filters to manage controller execution; implement asynchronous and synchronous actions; implement streaming actions	Added subtasks: Implement attribute routing; implement SignalR; test Web API web services
Secure a Web API Implement HTTPBasic authentication over SSL; implement Windows Auth; enable cross-domain requests; prevent cross-site request forgery (XSRF); implement, and extend, authorization filters to control access to the application	Removed subtask: Enable cross-domain requests Modified subtask: Design, implement, and extend authorization and authentication filters to control access to the application Added subtasks: Implement Cross Origin Request Sharing (CORS); Implement SSO by using OAuth 2.0; Configure multiple authentication modes on a single endpoint
Host and manage Web API Host Web API in an ASP.NET app; self-host a Web API in your own process (a Windows service); host services in a Windows Azure worker role; restricting message size; configure the host server for streaming	Modified subtask: Self-host a Web API in your own process (a Windows service) including Open Web Interface for .NET (OWIN)
Consume Web API web services Consume Web API services by using HttpClient synchronously and asynchronously; send and receive requests in different formats (JSON/HTML/etc.)	Added subtask: Request batching

5. Deploying web applications and services

Tasks currently measured	Tasks Added/Changed post *April 2014*
Design a deployment strategy Create an IIS install package; deploy to web farms; deploy a web application by using XCopy; automate a deployment from TFS or Build Server	No Change
Choose a deployment strategy for a Windows Azure web application Perform an in-place upgrade and VIP swap; configure an upgrade domain; create and configure input and internal endpoints; specify operating system configuration	Added subtask: Deploy applications using Azure Web Site
Configure a web application for deployment Switch from production/release mode to debug mode; use SetParameters to set up an IIS app pool, set permissions and passwords); configure WCF endpoints, bindings, and behaviors; transform web.config by using XSLT (for example, across development, test, and production/release environments); configure Azure configuration settings	Modified subtask: Configure WCF endpoints (including HTTPS protocol mapping), bindings, and behaviors Added subtask: Enable and monitor ASP.NET App Suspend
Manage packages by using NuGet Create and configure a NuGet package; install and update an existing NuGet package; connect to a local repository cache for NuGet, set up your own package repository	No Change
Create, configure, and publish a web package Create an IIS InstallPackage; configure the build process to output a web package; apply pre- and post- condition actions to ensure that transformations are correctly applied; include appropriate assets (web content, certificates)	Added subtask: Configure deployment
Share assemblies between multiple applications and servers Prepare the environment for use of assemblies across multiple servers (interning); sign assemblies by using a strong name; deploy assemblies to the global assembly cache; implement assembly versioning; create an assembly manifest; configure assembly binding redirects (for example, from MVC2 to MVC3)	Modified subtask: Configure assembly binding redirects (for example, from MVC4 to MVC5)