

## Microsoft Realtests 70-487 Questions & Answers

Number: 70-487  
Passing Score: 800  
Time Limit: 120 min  
File Version: 33.4



**Microsoft 70-487 Questions & Answers**

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

## General Exam

### QUESTION 1

You are building an ADO.NET Entity Framework application. You need to validate the conceptual schema definition language (CSDL), store schema definition language (SSDL), and mapping specification language (MSL) files. Which Entity Data Model tool can you use? (Each correct answer presents a complete solution.

Choose all that apply.)

- A. EDM Generator (EdmGen.exe)
- B. ADO.NET Entity Data Model Designer
- C. Entity Data Model Wizard
- D. Update Model Wizard

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 2

DRAG DROP

You are developing an ASP.NET Web API action method.

The action method must return the following JSON in the message body.

```
{ " Name ":" Fabrikam", "Vendor Id": 9823, "Items": ["Apples", "Oranges"] }
```

You need to return an anonymous object that is serialized to JSON.

What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Answer Area

"Fabrikam", VendorNumber = 9823,	<pre>public object Get() {     [ ]     {         Name = [ ]         Items = [ ]     }; }</pre>
"Fabrikam", VendorNumber = "9823",	
new List<string> { "Apples", "Oranges" }	
new List<string> { "Apples, Oranges" }	
return new List<string>	
return new	

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

```
public object Get()
{
    return new
    {
        Name = "Fabrikam", VendorNumber = 9823,
        Items = new List<string> { "Apples", "Oranges" }
    };
}
```

**QUESTION 3**

DRAG DROP

You are developing an ASP.NET MVC Web API image management application.

The application must meet the following requirements:

- It must send or receive image data without the use of a buffer.
- It must allow up to 4 MB of image data to be received.
- It must allow up to 3 MB of image data to be sent.

You need to complete the code to meet the requirements. What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

config

server

MaxBufferSize

MaxReceivedMessageSize

MaxConcurrentRequests

Streamed

Buffered

### Answer Area

```

class Program
{
    private static string _baseAddress = "http://localhost:8080/";

    static void Main(string[] args)
    {
        var config = new HttpSelfHostConfiguration(_baseAddress);
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );

        [ ] . [ ] = 1024 * 1024 * 3;

        [ ] . [ ] = 1024 * 1024 * 4;

        [ ] .TransferMode =

        TransferMode. [ ] ;

        var server = new HttpSelfHostServer(config);
        server.OpenAsync().Wait();
    }

```

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

```
class Program
{
    private static string _baseAddress = "http://localhost:8080/";

    static void Main(string[] args)
    {
        var config = new HttpSelfHostConfiguration(_baseAddress);
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}
```

```
config.MaxBufferSize = 1024 * 1024 * 3;
```

```
config.MaxReceivedMessageSize = 1024 * 1024 * 4;
```

```
config.TransferMode =
```

```
TransferMode.Streamed;
```

```
var server = new HttpSelfHostServer(config);
server.OpenAsync().Wait();
}
```

#### QUESTION 4

You are developing a library to support multiple ASP.NET MVC web applications on a shared server. The library provides implementations of security algorithms. If a problem with any of the security algorithms is discovered, a new version of the library must be created and deployed. Application downtime during the update must be minimized. You need to ensure that the new version of the library will be used by all applications as soon as possible. What should you do?

- A. Build the web applications and include the security assembly as an embedded resource.  
When an update is needed, copy the new assembly to the bin directory for the application.
- B. Sign all assemblies in each application with the same key used to sign the security assembly.  
When an update is needed, create a new key pair and re-sign all assemblies.

- C. Build the security assembly as a netmodule in a shared location.  
Use the assembly linker to merge the netmodule into the assemblies for the application.  
When an update is needed, update the netmodule in the shared location.
- D. Install the security assembly in the Global Assembly Cache (GAC).  
When an update is needed, update the assembly in the GAC.

**Correct Answer:** D

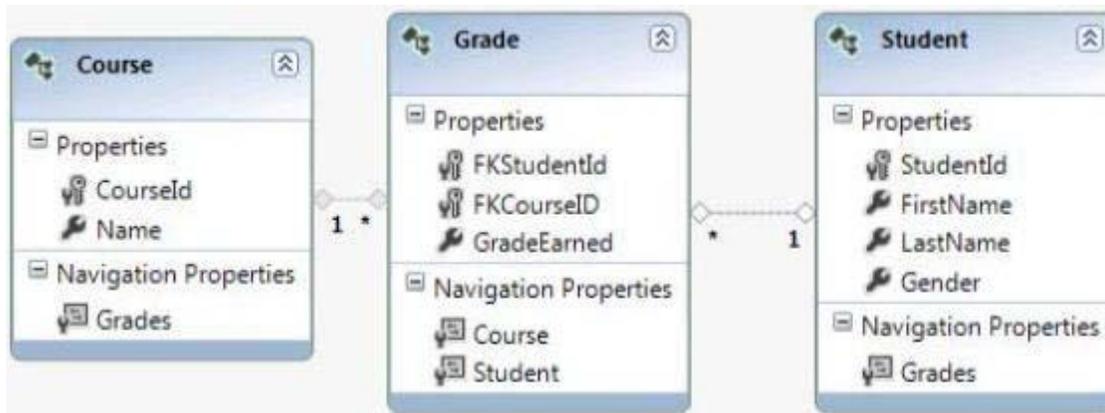
**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 5

You are developing an application in Visual Studio 2012 to display student information. The application contains the following Entity Framework model.



The application contains a WCF data service named DirectoryService.svc. You need to create a query expression to display all of the grades for students whose first name is "John". How should you build the expression?

- A. `http://localhost:54946/DirectoryService.svc/Students?$filter=FirstName eq 'John' &$expand=Grades`
- B. `http://localhost:54946/DirectoryService.svc/Students?$filter=FirstName eq 'John'/Grades`
- C. `http://localhost:54946/DirectoryService.svc/Students?$filter=FirstName = 'John' &$expand=Grades`
- D. `http://localhost:54946/DirectoryService.svc/Grades/Students?$filter=FirstName eq 'John'`

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 6**

**DRAG DROP**

You are developing a self-hosted WCF service that returns stock market information. The service must be discoverable by any client application. You need to build the service host. How should you build the host? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Answer Area

UdpDiscoveryEndpoint

DiscoveryEndpoint

ServiceBehaviorAttribute

ServiceDiscoveryBehavior

ServiceHost

```

static void Main(string[] args)
{
    Uri StockURI = new Uri("http://localhost:8733/StockTicker");
    var mytype = typeof(StockTickerService);

    using ( [input] host
           = new [input] (mytype, StockURI)
           {
               host.AddServiceEndpoint(typeof(IStockTickerService),
                                       new WSHttpBinding(), "");

               host.Description.Behaviors.Add(new [input] ());

               host.AddServiceEndpoint(new [input] ());

               host.Open();
               Console.ReadLine();
               host.Close();
           }
    )
    {
    }
}
                
```

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

```
static void Main(string[] args)
{
    Uri StockURI = new Uri("http://localhost:8733/StockTicker");
    var mytype = typeof(StockTickerService);

    using ( ServiceHost host

        = new ServiceHost (mytype, StockURI)

        {

            host.AddServiceEndpoint (typeof (IStockTickerService),
                new WSHttpBinding (), "");

            host.Description.Behaviors.Add (new ServiceDiscoveryBehavior ());

            host.AddServiceEndpoint (new DiscoveryEndpoint ());

            host.Open ();
            Console.ReadLine ();
            host.Close ();

        }

    }
}
```

#### QUESTION 7

You are developing a WCF service that compares several data sources. The service takes a long time to complete. The service must meet the following requirements:

- The client must be able to continue processing while the service is running.
- The service must initiate communication with the client application when processing is complete.

You need to choose a message pattern to meet the requirements. Which message pattern should you choose?

- A. One Way
- B. Streaming

- C. Duplex
- D. Request/Reply

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### **QUESTION 8**

**DRAG DROP**

You are developing a WCF service. You need to implement transport security by using NTLM authentication and NetTcpBindings. Which configuration values should you use? (To answer, drag the appropriate configuration values to the correct location or locations in the answer area. Each configuration value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

`binding="netTcpBinding"`

`binding="Duplex"`

`binding="NtlmTcp"`

`mode="netBindingTcp"`

`mode="Transport"`

`mode="Duplex"`

`clientCredentialType="netTcpBinding"`

`clientCredentialType="NtlmTcp"`

`clientCredentialType="Ntlm"`

Answer Area

```

<system.serviceModel>
  <protocolMapping>

    <add scheme="https" />

  </protocolMapping>
  <bindings>
    <wsHttpBinding>
      <binding>

        <security >

          <transport />

        </security>
      </binding>
    </wsHttpBinding>
  </bindings>
</system.serviceModel>

```

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

```
<system.serviceModel>
  <protocolMapping>

    <add scheme="https" binding="netTcpBinding" />

  </protocolMapping>
  <bindings>
    <wsHttpBinding>
      <binding>

        <security mode="Transport" />

        <transport clientCredentialType="Ntlm" />

      </security>
    </binding>
  </wsHttpBinding>
</bindings>
</system.serviceModel>
```

#### QUESTION 9

You are developing a WCF service. A new service instance must be created for each client request. You need to choose an instancing mode. Which instancing mode should you use?

- A. Single
- B. PerRequest
- C. PerCall
- D. Multiple
- E. PerSession

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 10**

**DRAG DROP**

You are developing a WCF service. You need to implement transport security by using NTLM authentication and NetTcpBindings. Which configuration values should you use? (To answer, drag the appropriate configuration values to the correct location or locations in the answer area. Each configuration value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

	Answer Area
<code>binding="netTcpBinding"</code>	<code>&lt;system.serviceModel&gt;</code>
<code>binding="Transport"</code>	<code>&lt;protocolMapping&gt;</code>
<code>binding="Ntlm"</code>	<code>&lt;add scheme="https" [ ] /&gt;</code>
<code>mode="netTcpBinding"</code>	<code>&lt;/protocolMapping&gt;</code>
<code>mode="Transport"</code>	<code>&lt;bindings&gt;</code>
<code>mode="Ntlm"</code>	<code>&lt;wsHttpBinding&gt;</code>
<code>clientCredentialType="netTcpBinding"</code>	<code>&lt;binding&gt;</code>
<code>clientCredentialType="Transport"</code>	<code>&lt;security [ ] &gt;</code>
<code>clientCredentialType="Ntlm"</code>	<code>&lt;transport [ ] /&gt;</code>
	<code>&lt;/security&gt;</code>
	<code>&lt;/binding&gt;</code>
	<code>&lt;/wsHttpBinding&gt;</code>
	<code>&lt;/bindings&gt;</code>
	<code>&lt;/system.serviceModel&gt;</code>

A.

**Correct Answer: A**

**Section: (none)**

## Explanation

### Explanation/Reference:

```
<system.serviceModel>
  <protocolMapping>

    <add scheme="https" binding="netTcpBinding" />

  </protocolMapping>
  <bindings>
    <wsHttpBinding>
      <binding>

        <security mode="Transport" >

          <transport clientCredentialType="Ntlm" />

        </security>
      </binding>
    </wsHttpBinding>
  </bindings>
</system.serviceModel>
```

## QUESTION 11

### DRAG DROP

You are developing an ASP.NET MVC Web API application. The application must meet the following requirements:

- It must send or receive data without the use of a buffer.
- It must allow up to 1 MB of data to be received.
- It must allow up to 2 MB of data to be sent.

You need to complete the code to meet the requirements. What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Answer Area

config

server

MaxBufferSize

MaxReceivedMessageSize

MaxConcurrentRequests

Streamed

Buffered

```

class Program
{
    private static string _baseAddress = "http://localhost:8080/";

    static void Main(string[] args)
    {
        var config = new HttpSelfHostConfiguration(_baseAddress);
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );

        [ ] . [ ] = 1024 * 1024 * 2;

        [ ] . [ ] = 1024 * 1024;

        [ ] .TransferMode =

        TransferMode. [ ] ;

        var server = new HttpSelfHostServer(config);
        server.OpenAsync().Wait();
    }
}
                
```

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

```
class Program
{
    private static string _baseAddress = "http://localhost:8080/";

    static void Main(string[] args)
    {
        var config = new HttpSelfHostConfiguration(_baseAddress);
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}
```

config

.MaxBufferSize

= 1024 \* 1024 \* 2;

config

.MaxReceivedMessageSize

= 1024 \* 1024;

config

.TransferMode =

TransferMode.Streamed ;

```
var server = new HttpSelfHostServer(config);
server.OpenAsync().Wait();
```

```
}
}
```

**QUESTION 12**

**DRAG DROP**

You are developing an ASP.NET Web API action method.

The action method must return the following JSON in the message body.

```
{"Name": "Fabrikam", "VendorId": 9823, "Items": ["Dogs", "Cats"]} >
```

You need to return an anonymous object that is serialized to JSON.

What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Answer Area

"Fabrikam", VendorNumber = 9823,

"Fabrikam", VendorNumber = "9823",

new List<string> { "Dogs", "Cats" }

new List<string> { "Dogs, Cats" }

return new List<string>

return new

```

public object Get()
{
    [ ]
    {
        Name = [ ]
        Items = [ ]
    }
};
        
```

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

```

public object Get()
{
    return new [ ]
    {
        Name = "Fabrikam", VendorNumber = 9823, [ ]
        Items = new List<string> { "Dogs", "Cats" } [ ]
    }
};
    
```

**QUESTION 13**  
DRAG DROP

You are developing an ASP.NET Web API application for currency conversion that will be consumed by a web browser by using a composite application that is served from another web domain. You need to configure the Web API. What should you do? (To answer, drag the appropriate XML elements to the correct location or locations in the answer area. Each XML element may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

	Answer Area
Access-Control-Allow-Origin	<pre> &lt;httpProtocol&gt;   &lt;customHeaders&gt;     &lt;add name="Access-Control-Allow-Origin"       value=" " /&gt;     &lt;add name=" "       value="PUT, DELETE"/&gt;     &lt;add name=" "       value=" " /&gt;   &lt;/customHeaders&gt; &lt;/httpProtocol&gt; </pre>
Access-Control-Allow-Headers	
Access-Control-Allow-Methods	
Access-Control-Allow-Request-Method	
Access-Control-Allow-Request-Headers	
*	
POST, GET	
Content-Type	

A.

**Correct Answer:** A  
**Section:** (none)  
**Explanation**

**Explanation/Reference:**

```
<httpProtocol>
  <customHeaders>
    <add name="Access-Control-Allow-Origin"
      value="*" />
    <add name="Access-Control-Allow-Methods"
      value="PUT, DELETE" />
    <add name="Access-Control-Allow-Headers"
      value="Content-Type" />
  </customHeaders>
</httpProtocol>
```

#### QUESTION 14

You are developing an ASP.NET MVC application. The application is an order processing system that uses the ADO.NET Entity Framework against a SQL Server database. It has a controller that loads a page that displays all orders along with customer information. Lazy loading has been disabled. The Order class is shown below.

```
public partial class Order
{
    ...
    public string CustomerID { get; set; }
    ...
    public virtual Customer Customer { get; set; }
}
```

You need to return the orders and customer information in a single round trip to the database. Which code segment should you use?

- A. 

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders;
    orders = orders.Include("Customer");
    return View(orders.ToList());
}
```
- B. 

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders.Include("Order.Customer");
    return View(orders.ToList());
}
```
- C. 

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders;
    orders.Select(o => o.Customer).Load();
    return View(orders.ToList());
}
```
- D. 

```
public ActionResult Index()
{
    IQueryable<Order> orders = db.Orders;
    return View(orders.ToList());
}
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 15**

You are developing an ASP.NET MVC application that reads and writes data from a SQL Server database. You need to maintain data integrity in all situations that use transactions.



- A. ReadUncommitted
- B. Repeatable
- C. Serializable
- D. ReadCommitted

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 16**

You are developing an ASP.NET MVC application. Deployment administrators do not have access to Visual Studio 2102, but will have the elevated permissions required to deploy the application to the servers. You need to select a deployment tool for use by the deployment administrators. Which tool should you use?

- A. Publish Web Site Tool
- B. Web Deployment Package
- C. One-Click Publish
- D. Deployment Package Editor

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

## Case 1

### QUESTION 1

Case Study: 1

#### Scenario 1

#### Background

You are developing a flight information consolidation service. The service retrieves flight information from a number of sources and combines them into a single data set. The consolidated flight information is stored in a SQL Server database. Customers can query and retrieve the data by using a REST API provided by the service. The service also offers access to historical flight information. The historical flight information can be filtered and queried in an ad hoc manner. The service runs on a Windows Azure Web Role. SSL is not used.

#### Business Requirements

- A new data source for historical flight information is being developed by a contractor located on another continent.
- If a time zone is not specified, then it should be interpreted as Coordinated Universal Time (UTC).
- When you upgrade a service from a staging deployment to a production deployment, the time that the service is unavailable must be minimized.
- The default port must be used for HTTP.

#### Technical Requirements

The existing sources of flight information and the mechanism of exchange are listed below.

- Blue Yonder Airlines provides flight information in an XML file.
- Consolidated Messenger provides flight information in a Microsoft Access database that is uploaded every 12 hours to the service using SFTP. The company uses port 22 for SFTP.
- Margie's Travel provides and consumes flight information using serialized ADO.NET DataSets. Data is periodically synced between the service and Margie's Travel.
- Trey Research provides data from multiple sources serialized in proprietary binary formats. The data must be read by using .NET assemblies provided by Trey Research. The assemblies use a common set of dependencies. The current version of the Trey Research assemblies is 1.2.0.0. All assemblies provided by Trey Research are signed with a key pair contained in a file named Trey.snk, which Trey Research also supplies.
- The application specification requires that any third-party assemblies must have strong names.

#### Application Structure

### FlightInfo.cs

```
public class FlightInfo
{
    string DataSource { get; set; }
    public string Airline { get; set; }
    public string Flight { get; set; }
    public DateTimeOffset Arrival { get; set; }
    public int Seats { get; set; }
    public bool WasLate { get; set; }
}
```

### BlueYonderLoader.cs

```
public class BlueYonderLoader
{
    public IEnumerable<RawFlightData> LoadFlights(XDocument feed)
    {
        ...
    }

    private RawFlightData Parse(XElement flightElement)
    {
        ...
    }
}
```

## HistoricalDataLoader.cs

```
public class HistoricalDataLoader
{
    public static IEnumerable<HistoricalFlightInfo> LoadHistoricalFlights()
    {
        ...
    }

    public void StreamHistoricalFlights(XmlWriter responseWriter, string airline)
    {
        ...
    }

    private XElement ConvertToHistoricalFlight(XElement flight)
    {
        return new XElement("Flight", flight);
    }

    private string GetAirline(XElement flightName)
    {
        return flightName.Value.Substring(0, 2);
    }

    IEnumerable<XElement> RemoteDataStream()
    {
        return XDocument.Load("").Elements();
    }
}
```

### MargiesTravelSync.cs

```
public class MargiesTravelSync
{
    public void Sync()
    {
        ...
    }

    private DataSet LoadLocal()
    {
        var dataSet = new DataSet();
        dataSet.ReadXml("local");
        return dataSet;
    }

    private StreamWriter SendStream()
    {
        return new StreamWriter("SendStream");
    }

    private StreamReader ReceiveStream()
    {
        return new StreamReader("ReceiveStream");
    }
}
```



### FlightInfoContext.cs

```
public class FlightInfoContext : DbContext
{
    public DbSet<FlightInfo> FlightInfo { get; set; }

    public override int SaveChanges()
    {
        return base.SaveChanges();
    }

    private bool IsTransient(int ex)
    {
        var errors = new[] { 10053, 10054, 64 };
        return errors.Contains(ex);
    }
}
```

### FlightDataController.cs

```
public class FlightDataController : ApiController
{
    FlightInfoContext _Context;

    public FlightDataController()
    {
        _Context = new FlightInfoContext();
    }

    [HttpGet]
    public IEnumerable<FlightInfo> GetFlightInfo()
    {
        return _Context.FlightInfo.Select(x => x).AsEnumerable();
    }

    private IEnumerable<HistoricalFlightInfo> LoadHistorical()
    {
        return HistoricalDataLoader.LoadHistoricalFlights();
    }
}
```

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 2

You need to recommend a data access technology to the contractor to retrieve data from the new data source. Which data access technology should you recommend?

- A. LINQ to XML
- B. ADO.NET Entity Framework
- C. ADO.NET DataSets
- D. WCF Data Services

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 3

Data provided by Consolidated Messenger is cached in the HttpContext.Cache object. You need to ensure that the cache is correctly updated when new data arrives. What should you do?

- A. Ensure that the EffectivePrivateBytesLimit value is greater than the size of the database file.
- B. Change the sliding expiration of the cache item to 12 hours.
- C. Use the SqlCacheDependency type configured with a connection string to the database file.
- D. Use the CacheDependency type configured to monitor the SFTP target folder.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 4**  
DRAG DROP

You need to parse flight information from Blue Yonder Airlines. The content of the XML file is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<AirlineFeed>
  <Flight xmlns="urn:CFI" name="AS515">
    <Seats>123</Seats>
    <Arrival>5/2/2011 12:01:13</Arrival>
  </Flight>
  <Flight name="UN24">
    <Seats>123</Seats>
    <Arrival>5/1/2012 10:17:57 PM +02:00</Arrival>
  </Flight>
  <FlightManifest>
    ...
  </FlightManifest>
</AirlineFeed>
```

Some airlines do not specify the timezone of the arrival time. If the timezone is not specified, then it should be interpreted per the business requirements. You need to implement the LoadFlights() and Parse() methods of the BlueYonderLoader class. What should you do? (To answer, drag the appropriate code segments to the correct location in the answer area. Each segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

```
var flights = feed.Elements(  
    feed.Root.GetPrefixOfNamespace("{urn:CFI}") + "Flight");
```

```
var flights = feed.Descendants().Where(x =>  
    x.NodeType != XmlNodeType.XmlDeclaration && (string)x ==  
    "Flight");
```

```
var flights = feed.Descendants("{urn:CFI}Flight")  
    .Concat(feed.Descendants("Flight"));
```

```
fi.Arrival = DateTimeOffset.Parse(arrivalRaw,  
    null, System.Globalization.DateTimeStyles.AssumeUniversal);
```

```
fi.Arrival = DateTimeOffset.Parse(arrivalRaw,  
    null, System.Globalization.DateTimeStyles.AdjustToUniversal);
```

```
fi.Arrival = XmlConvert.ToDateTimeOffset(arrivalRaw,  
    new[] { "Local", "Universal" });
```

```
public IEnumerable<FlightInfo> LoadFlights(XDocument feed)
```

```
{
```

```
    return flights.Select(x => Parse(x));
```

```
}
```

```
private FlightInfo Parse(XElement flightElement)
```

```
{
```

```
    var fi = new FlightInfo();
```

```
    fi.Flight = flightElement.Attribute("name").Value;
```

```
    var arrivalRaw = flightElement.Element("Arrival").Value;
```

A.

**Correct Answer:**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

```
public IEnumerable<FlightInfo> LoadFlights(XDocument feed)
{
    var flights = feed.Descendants("{urn:CFI}Flight")
        .Concat(feed.Descendants("Flight"));

    return flights.Select(x => Parse(x));
}

private FlightInfo Parse(XElement flightElement)
{
    var fi = new FlightInfo();
    fi.Flight = flightElement.Attribute("name").Value;
    var arrivalRaw = flightElement.Element("Arrival").Value;

    fi.Arrival = DateTimeOffset.Parse(arrivalRaw,
        null, System.Globalization.DateTimeStyles.AssumeUniversal);

    fi.Seats = XmlConvert.ToInt32(flightElement.Element("Seats").Value);
    return fi;
}
```

### QUESTION 5

Historical flight information data will be stored in Windows Azure Table Storage using the FlightInfo class as the table entity. There are millions of entries in the table. Queries for historical flight information specify a set of airlines to search and whether the query should return only late flights. Results should be ordered by flight name. You need to specify which properties of the FlightInfo class should be used at the partition and row keys to ensure that query results are returned as quickly as possible. What should you do? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Use the WasLate property as the row key.
- B. Use the Airline property as the row key.

- C. Use the WasLate property as the partition key
- D. Use the Arrival property as the row key.
- E. Use the Airline property as the partition key.
- F. Use the Flight property as the row key.

**Correct Answer:** CF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 6**

Errors occasionally occur when saving data using the FlightInfoContext ADO.NET Entity Framework context. Updates to the data are being lost when an error occurs. You need to ensure that data is still saved when an error occurs by retrying the operation. No more than five retries should be performed. With which code segment should you replace the body of the SaveChanges() method in the FlightInfoContext.cs file?

- A. 

```
var result = FlightInfo.SqlQuery("UPDATE WITH RETRY", FlightInfo, "IsTransient", 5);
if (result.Count() > 5)
{
    result.AsNoTracking();
    return -1;
}
return 0;
```
- B. 

```
try
{
    return base.SaveChanges();
}
catch (EntityCommandExecutionException ex)
{
    if (ex.Data.Keys.Cast<int>().Any(x => IsTransient(x)))
    {
        return 5 & SaveChanges();
    }
    return -1;
}
```
- C. 

```
for (var i = 0; i < 5; i++)
{
    try
    {
        return base.SaveChanges();
    }
    catch (SqlException ex)
    {
        if (IsTransient(ex.Number))
        {
            continue;
        }
    }
}
return base.SaveChanges();
```
- D. 

```
var exception = new EntitySqlException();
while (exception.HResult != 0 && exception.Data.Count < 5)
{
    try
    {
        return base.SaveChanges();
    }
    catch (EntitySqlException ex)
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

EntitySQLException : Represents errors that occur when parsing Entity SQL command text. This exception is thrown when syntactic or semantic rules are violated.

SQLException : The exception that is thrown when SQL Server returns a warning or error. This class cannot be inherited.

EntityCommandExecutionException : Represents errors that occur when the underlying storage provider could not execute the specified command. This exception usually wraps a provider-specific exception.

## Case 2

### QUESTION 1

Case Study: 2

Scenario 2

Background

You are developing an ASP.NET MVC application in Visual Studio 2012 that will be used to process orders.

Business Requirements

The application contains the following three pages.

- A page that queries an external database for orders that are ready to be processed. The user can then process the order.
- A page to view processed orders.
- A page to view vendor information.

The application consumes three WCF services to retrieve external data.

Technical Requirements

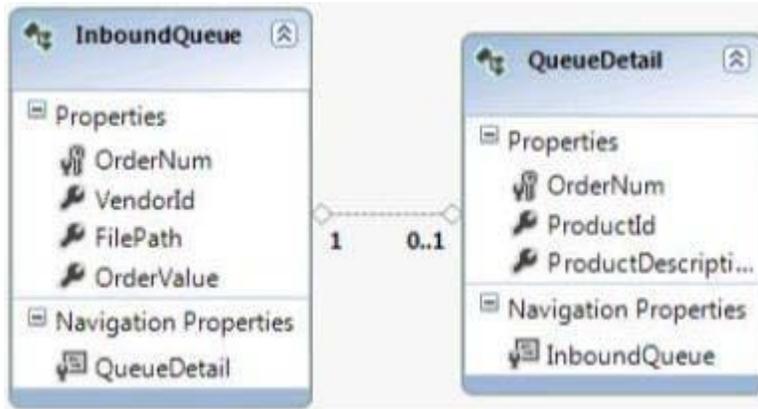
Visual Studio Solution:

The solution contains the following four projects.

- ExternalQueue: A WCF service project used to communicate with the external order database.
- OrderProcessor: An ASP.NET MVC project used for order processing and logging order metadata.
- OrderUpload: A WCF service project used to submit order data to an external data source.
- Shipping: A WCF service project used to acquire shipping information.

ExternalQueue Project:

Entity Framework is used for data access. The entities are defined in the ExternalOrders.edmx file as shown in the following diagram.



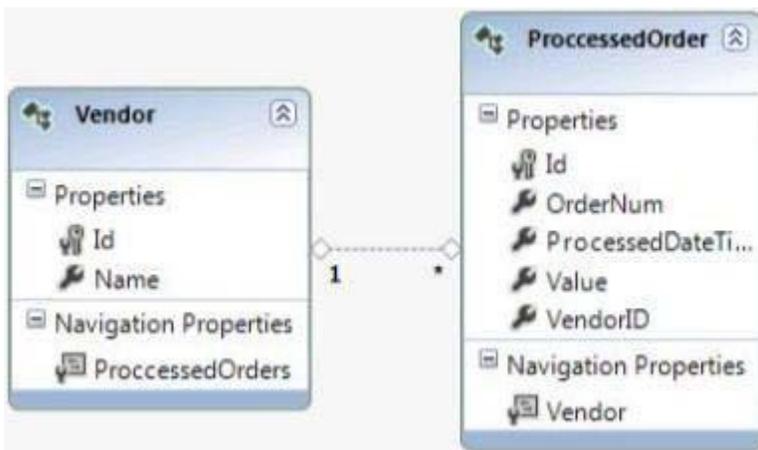
The project contains two services defined in the following files.

- IExternalQueueService.cs
- ExternalQueueService.svc.

The ExternalQueue.Helpers namespace contains a definition for a class named OrderNotFound Exception.

OrderProcessor Project:

Entity Framework is used for data access. The entities are defined in the ProcessedOrders.edmx file as shown in the following diagram.



The classes are contained in the OrderProcessor.Entities namespace. The project contains the following two controllers.

- InboundQueueController.cs
- ProcessedOrderController.cs

WCF service proxies to the ExternalQueue, Shipping and OrderUpload services have been generated by using the command prompt. The ExecuteCommandProcedure() method in the ExternalQueueService.svc file must run asynchronously.

The ProcessedOrderController controller has the following requirements.

The GetVendorPolicy() method must enforce a 10 minute absolute cache expiration policy.

The GetProcessedOrders() method must return a view of the 10 most recently processed orders.

OrderUpload Project:

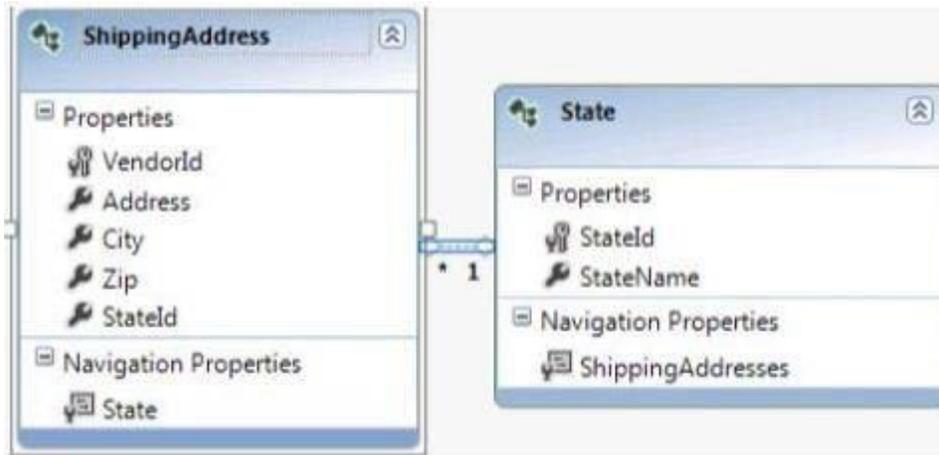
The project contains two services defined in the following files.

- IUploadCallbackService.cs
- UploadCallbackService.svc

Data Access is maintained in a file named UploadOrder.es.

Shipping Project:

Entity Framework is used for data access. The entities are defined in the ExternalOrders.edmx file as shown in the following diagram.



The Custom Tool property for ExternalOrders.edmx has been removed. POCO classes for the Entity Model are located in the ShippingAddress.cs file. The POCO entity must be loaded by using lazy loading. The project contains two services defined in the following files.

- IShippingService.cs
- ShippingService.svc.

The IShippingService contract must contain an operation that receives an order number as a parameter. The operation must return a class named ShippingInfo that inherits from a class named State.

Application Structure

ExternalQueue\IExternalQueueService.cs

```
IQ01 using System.Collections.Generic;
IQ02 using System.ServiceModel;
IQ03 using ExternalQueue.Helpers;
IQ04
IQ05 namespace ExternalQueue
IQ06 {
IQ07     [ServiceContract]
IQ08     public interface IExternalQueueService
IQ09     {
IQ10         [OperationContract]
IQ11         List<Entities.InboundQueue> GetExternalOrders();
IQ12
IQ13         [FaultContract(typeof(OrderNotFoundException))]
IQ14         [OperationContract]
IQ15         void DeleteExternalOrder(int orderNum);
IQ16
IQ17         [OperationContract]
IQ18         Entities.InboundQueue GetExternalOrder(int orderNum);
IQ19     }
IQ20 }
```

OrderProcessor\IExternalQueueService.svc

```
EQ01 using System;
EQ02 using System.Collections.Generic;
EQ03 using System.Linq;
EQ04 using System.Data.EntityClient;
EQ05 using System.Data;
EQ06 using ExternalQueue.Entities;
EQ07 using System.Data.Objects;
EQ08 using ExternalQueue.Helpers;
EQ09 using System.ServiceModel;
EQ10 using System.Threading.Tasks;
EQ11
EQ12 namespace ExternalQueue
EQ13 {
EQ14     public class ExternalQueueService : IExternalQueueService
EQ15     {
EQ16         public List<Entities.InboundQueue> GetExternalOrders()
EQ17         {
EQ18             List<InboundQueue> queueItems = new List<InboundQueue>();
EQ19             return queueItems;
EQ20         }
EQ21
EQ22         public void DeleteExternalOrder(int orderNum)
EQ23         {
EQ24             using (var context = new ExternalOrdersEntities())
EQ25             {
EQ26                 var orders = context.InboundQueues.Where(i => i.OrderNum ==
EQ27 orderNum).ToList();
EQ28                 if (orders.Count() > 0)
EQ29                 {
EQ30                     using (EntityCommand cmd = new EntityCommand())
EQ31                     {
EQ32                         cmd.CommandText = "ExternalOrdersEntities.uspInboundQueueDelete";
EQ33                         cmd.CommandType = CommandType.StoredProcedure;
EQ34                         EntityParameter param = new EntityParameter();
EQ35                         param.Value = orderNum;
EQ36                         param.ParameterName = "orderNum";
EQ37                         cmd.Parameters.Add(param);
EQ38                         ExecuteCommandProcedure(cmd);
EQ39                     }
EQ40                 }
EQ41                 else
EQ42                 {
EQ43                     OrderNotFoundException ex = new OrderNotFoundException();
EQ44                     ex.OrderNum = orderNum;
EQ45                     ex.ExceptionMessage = "Order not found...Cannot delete";
EQ46                 }
EQ47             }

```

ExternalQueue\ProcessedOrderController.cs

```
PC01 using System;
PC02 using System.Collections.Generic;
PC03 using System.Linq;
PC04 using System.Runtime.Caching;
PC05 using System.Web.Mvc;
PC06 using OrderProcessor.Entities;
PC07 using OrderProcessor.Helpers;
PC08 using System.Configuration;
PC09
PC10 namespace OrderProcessor.Controllers
PC11 {
PC12     public class ProcessedOrderController : Controller
PC13     {
PC14         public ActionResult GetProcessedOrders()
PC15         {
PC16             using (var context = new ProcessedOrders())
PC17             {
PC18                 List<Entities.ProcessedOrder> orders = new List<ProcessedOrder>();
PC19                 return View(orders);
PC20             }
PC21         }
PC22
PC23         private ObjectCache cache {get { return MemoryCache.Default; }}
PC24
PC25         public ActionResult GetVendors()
PC26         {
PC27             List<Entities.Vendor> vendors = cache.Get
("vendorKey") as List<Entities.Vendor>;
PC28             if (vendors == null)
PC29             {
PC30                 using (var context = new ProcessedOrders())
PC31                 {
PC32                     vendors = context.Vendors.ToList();
PC33                 }
PC34             }
PC35             return View(vendors);
PC36         }
PC37     }
PC38
PC39     private CacheItemPolicy GetVendorPolicy()
PC40     {
PC41         CacheItemPolicy vendorPolicy = new CacheItemPolicy();
PC42     }
```

OrderProcessor\InboundQueueController.cs

```
IC01 using System;
IC02 using System.Collections.Generic;
IC03 using System.Web.Mvc;
IC04 using OrderProcessor.Entities;
IC05 using ExternalQueue.Entities;
IC06 using System.ServiceModel;
IC07 using System.Collections;
IC08 using ExternalQueue.Helpers;
IC09 using OrderProcessor.Helpers;
IC10 using System.Linq;
IC11
IC12 namespace OrderProcessor.Controllers
IC13 {
IC14     public class InboundQueueController : Controller
IC15     {
IC16         public ActionResult GetQueueItems()
IC17         {
IC18             IEnumerable<InboundQueue> inboundOrders = Enumerable.Empty<InboundQueue>();
IC19             return View(inboundOrders);
IC20         }
IC21
IC22         public ActionResult ProcessOrder(int orderNum)
IC23         {
IC24             ExternalQueueServiceClient qService = new ExternalQueueServiceClient();
IC25             InboundQueue externalOrder = qService.GetExternalOrder(orderNum);
IC26             if (externalOrder != null)
IC27             {
IC28                 using (var context = new ProcessedOrders())
IC29                 {
IC30                     ProcessedOrder order = new ProcessedOrder();
IC31                     order.OrderNum = externalOrder.OrderNum;
IC32                     order.Value = Convert.ToDouble(externalOrder.OrderValue);
IC33                     order.VendorID = Convert.ToInt32(externalOrder.VendorId);
IC34                     order.ProcessedDateTime = DateTime.Now;
IC35                     context.ProcessedOrders.Add(order);
IC36                     context.SaveChanges();
IC37                 }
IC38                 qService.DeleteExternalOrder(orderNum);
IC39             }
IC40             return RedirectToAction("GetQueueItems");
IC41         }
IC42
IC43         public ActionResult ViewShippingInfo(int orderNum)
IC44         {
IC45             ShippingServiceClient shipService = new ShippingServiceClient();
IC46             var info = shipService.GetShippingInfo(orderNum);
```

OrderUpload\IUploadCallbackService.cs

```
IU01 using System.ServiceModel;
IU02
IU03 namespace OrderUpload
IU04 {
IU05     [ServiceContract(CallbackContract = typeof(IUploadCallback))]
IU06     public interface IUploadCallbackService
IU07     {
IU08         [OperationContract]
IU09         void UploadOrder(int orderNum);
IU10     }
IU11
IU12     public interface IUploadCallback
IU13     {
IU14         [OperationContract]
IU15         decimal GetOrderValue(int orderNum);
IU16     }
IU17 }
```

OrderUpload\UploadCallbackService.svc

```
US01 using System.ServiceModel;
US02
US03 namespace OrderUpload
US04 {
US05     public class UploadCallbackService : IUploadCallbackService
US06     {
US07         public void UploadOrder(int orderNum)
US08         {
US09         }
US10     }
US11 }
```

Shipping\IShippingService.cs

```
IS01 using System.Runtime.Serialization;
IS02 using System.ServiceModel;
IS03
IS04 namespace Shipping
IS05 {
IS06     public interface IShippingService
IS07     {
IS08     }
IS09 }
IS10 }
```

Shipping\ShippingAddress.cs

```
SA01 using System.Collections.Generic;
SA02 using System.Data.Objects;
SA03
SA04 namespace Shipping.POCO
SA05 {
SA06     public class ShippingAddress
SA07     {
SA08         public int VendorId { get; set; }
SA09         public string Address { get; set; }
SA10         public string City { get; set; }
SA11         public int StateId { get; set; }
SA12         public string Zip { get; set; }
SA13         public State State { get; set; }
SA14     }
SA15
SA16     public class State
SA17     {
SA18         public int StateId { get; set; }
SA19         public string StateName { get; set; }
SA20         public List<ShippingAddress> ShippingAddresses { get; set; }
SA21     }
SA22 }
```

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

## QUESTION 2

DRAG DROP

You add a class named ShippingInfo. You need to modify the IShippingService interface and the ShippingInfo class to meet the technical requirements. What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)



Answer Area

[DataMember]

[CollectionDataContract]

[DataContract]

[ServiceContract]

[OperationContract]

```

public interface IShippingService
{
    ShippingInfo GetShippingInfo(int orderNum);
}

public class State
{
    public string StateName { get; set; }
}

public class ShippingInfo : State
{
    public string StreetAddress { get; set; }

    public string ZipCode { get; set; }
}
    
```

A.

Correct Answer: A  
Section: (none)  
Explanation

Explanation/Reference:

```
[OperationContract]
public interface IShippingService
{
    [OperationContract]
    ShippingInfo GetShippingInfo(int orderNum);
}

[DataContract]
public class State
{
    [DataMember]
    public string StateName { get; set; }
}

[DataContract]
public class ShippingInfo : State
{
    [DataMember]
    public string StreetAddress { get; set; }

    [DataMember]
    public string ZipCode { get; set; }
}
```

### QUESTION 3

DRAG DROP

You need to modify the ExecuteCommandProcedure() method to meet the technical requirements. Which code segment should you use?

	Answer Area
<code>await connection.OpenAsync();</code>	<pre>private async Task ExecuteCommandProcedure(EntityCommand command) {     using (EntityConnection connection         = new EntityConnection("name=ExternalOrdersEntities"))     {         command.Connection = connection:         <input type="text"/>         <input type="text"/>     } }</pre>
<code>await command.ExecuteNonQueryAsync();</code>	
<code>connection.OpenAsync();</code>	
<code>command.OpenAsync();</code>	
<code>await command.QueryAsync();</code>	

A.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Second answer is wrong, it should be "await command.ExecuteNonQuery();" "

```
private async Task ExecuteCommandProcedure(EntityCommand command)
{
    using (EntityConnection connection
        = new EntityConnection("name=ExternalOrdersEntities"))
    {
        command.Connection = connection;

        await connection.OpenAsync();

        await command.QueryAsync();
    }
}
```

Credits to Rem

#### QUESTION 4

The GetVendors() action in the ProcessedOrderController controller is querying the database each time it is run. The GetVendors() action must query the database only if the cache is null.

You need to add code to the action at line PC33 to cache the data. Which code segment can you use? (Each correct answer presents a complete solution. Choose all that apply.)

- A. cache.Set(new CacheItem("vendorKey", vendors), GetVendorPolicy());
- B. cache.Add("vendors", vendors, new CacheItemPolicy());
- C. cache.Add(new CacheItem("vendorKey", vendors) , GetVendorPolicy());
- D. cache.AddOrGetExisting("vendorKey", context, new CacheItemPolicy());

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 5

DRAG DROP

The UploadOrder() method in the UploadCallbackService service is not implementing the callback behavior defined in the IUploadCallBackService

interface. You need to modify the class to implement the required callback behavior. What should you do? (To answer, drag the appropriate code segments to the



correct location or locations in the answer area. Each code segments may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Multiple

Single

GetOrderValue

UploadCallbackService

IUploadCallback

Answer Area

```

[ServiceBehavior(ConcurrencyMode =
    ConcurrencyMode. [ ] )]

public class UploadCallbackService : IUploadCallbackService
{
    public void UploadOrder(int orderNum)
    {
        [ ] callback = OperationContext
            .Current.GetCallbackChannel< [ ] >();
        decimal value = callback. [ ] (orderNum);

        UploadDB.UploadOrder.Upload(orderNum, value);
    }
}
                
```

A.

**Correct Answer:** A  
**Section:** (none)  
**Explanation**

**Explanation/Reference:**

```
[ServiceBehavior(ConcurrencyMode =  
    ConcurrencyMode.Single )]  
  
public class UploadCallbackService : IUploadCallbackService  
{  
    public void UploadOrder(int orderNum)  
    {  
        IUploadCallback callback = OperationContext  
            .Current.GetCallbackChannel< IUploadCallback >();  
        decimal value = callback.GetOrderValue(orderNum);  
  
        UploadDB.UploadOrder.Upload(orderNum, value);  
    }  
}
```