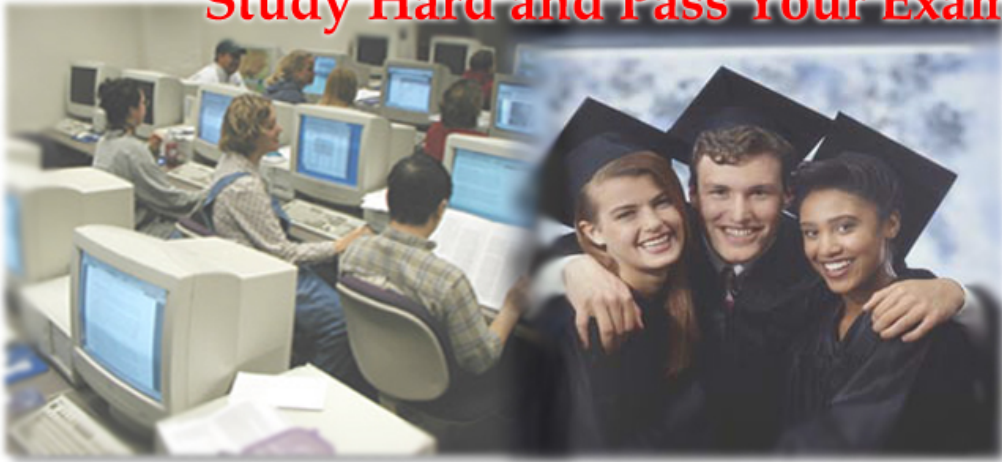


CertGuaranteed

Study Hard and Pass Your Exam



Exam : 070-529

Title : Microsoft .NET Framework 2.0 - Distributed
Application Development

Ver : 02.14.07

QUESTION 1

You have a Web service that is deployed on an unsecured network. You need to implement encryption on the Web service. The implementation must be configurable at run time. What should you do?

- A. Apply Web Services Enhancements (WSE) 3.0 security to the Web service that is configured to use an X.509 certificate with the Sign-Only protection level.
- B. Write custom code in each Web method of the Web service that encrypts the data by using the DESCryptoServiceProvider class.
- C. Write a custom SOAP extension attribute class to encrypt and decrypt the SOAP message. Apply the attribute to the Web service class.
- D. Apply Web Services Enhancements (WSE) 3.0 security to the Web service that is configured to use an X.509 certificate with the Sign and Encrypt protection level.

Answer: D

Explanation:

Only by using an X.509 certificate with Sign and Encrypt protection (D) can the parties of the message ensure that the message has not only not been tampered with, but that it can only be read by parties with the appropriate certificate.

Encryption on Web Services that is configurable at runtime is done through Web Services Enhancements (WSE) 3.0 security in 2.0, thus writing custom code (B) or using a custom SOAP extension (C) is not the most efficient or correct answer.

Using an X.509 certificate with Sign-Only protection (A), makes a hash of the content, and ensures at the other end that the message has not been tampered with - but does not ensure that no-one has read it along the way.

QUESTION 2

You create .NET Framework remoting components that must be deployed on an unattended application server. IIS is not installed on the application server. Company policy requires that no changes be made to the application server except the deployment of approved custom code that is written by your development team. You need to ensure that the components can run on the unattended application server.

What should you do?

- A. Create an ASP.NET Web service Web site to host the components.
- B. Create a Console Application project to host the components.
- C. Create a Windows-based application project to host the components.
- D. Create an ASP.NET Web application Web site to host the components.
- E. Create a Windows service to host the components.

Answer: E

Explanation:

Windows Services (E) are primarily designed for running unattended code without the

need for user interaction, and is the correct answer.

ASP .NET Web Services (A) require IIS in order to run. The application server does not have IIS installed.

Console Applications (B) are run from the command line with input and output information being exchanged between the command prompt and the running application. This is an unattended application server.

Windows-based applications (C) are graphical interface programs that require some user interaction. This is an unattended application server.

ASP .NET Web applications (D) require IIS, and are also provide a visual interface for users. This application neither has IIS installed, nor is it attended.

QUESTION 3

An application contains .NET Framework Web services that are built by using Microsoft Visual Studio 2005. Discovery information for these Web services must be made available to client applications. You need to expose the Disco files for all Web services in the application. What should you do?

- A. Use the Sproxy.exe tool to create the Disco files from the .wsdl files.
- B. Add the description parameter to the WebService attribute in the class files.
- C. Add a .vsdisco file to the application root and add the corresponding httpHandler to the Web.config file.
- D. Add a set of Disco files to the application root and add the PageHandlerFactory httpHandler to the Web.config file.

Answer: C

Explanation:

By placing a .vsdisco file in the virtual root (C) it is handled by the DiscoveryRequestHandler, which dynamically generates a DISCO document based on the resources found in the vroot, thus exposing all Web Services in the application. The SProxy.exe utility (A) is a command-line tool used to create a Web Services proxy class based on the WSDL description, it does not create discovery documents. The Description Property of the WebService attribute (B) provides a descriptive message for the XML Web Service.

While it is possible to use DISCO files in the virtual root to discover the capabilities of each web service (through documentation) and how to interact with it (through WSDL) (D), the PageHandlerFactory httpHandler finds the class that represents the ASP.NET web page that is being requested, and has nothing to do with discovery.

QUESTION 4

A Web Services Enhancements (WSE) 3.0 router application uses a referral cache to make routing decisions. When a client application sends a SOAP message to the WSE router application, the following exception is thrown.

Microsoft.Web.Services3.Addressing.AddressingFault: MessageInformation Header RequiredThe referral cache used by the WSE router application is as follows:

```
<?xml version="1.0" ?><r:referrals
```

```
xmlns:r="http://schemas.xmlsoap.org/ws/2001/10/referral"> <r:ref>
<r:for>
<r:exact>
http://www.contoso.com/WSERouter/Service.asmx
</r:exact>
</r:for>
<r:if />
<r:go>
<r:via>http://www.contoso.com/WSEService/Service.asmx</r:via>
</r:go>
<r:refId>uuid:fa469956-0057-4e77-962a-81c5e292f2ae</r:refId> </r:ref>
</r:referrals>
```

You need to ensure that a SOAP message can be sent to the WSE router application without throwing the AddressingFault exception. What should you do?

- A. Set the ConnectionGroupName property on the Web service client proxy to Allow Routing.
- B. Set the UserAgent property on the Web service client proxy to WSE 3.0.
- C. Set the AllowAutoRedirect property on the Web service client proxy to True.
- D. Change the Web service client proxy class definition so that it is derived from the Microsoft.Web.Services3.WebServicesClientProtocol class.

Answer: D

Explanation:

By default, when a proxy class is created for a client application it derives from the SoapHttpClientProtocol class. To use the functionality of WSE, the class that the proxy class derives from must be WebServicesClientProtocol (D)

The other three properties, ConnectionGroupName (A), UserAgent (B), and AllowAutoRedirect (C) inherit from the SoapHttpClientProtocol class and are not used with WSE 3.0.

QUESTION 5

You write a Web service that uses Web Services Enhancements (WSE) 3.0 for security. The Web service must support both the kerberosSecurity policy and the mutualCertificateSecurity policy. Both policies have the same options set for encryption and signatures. You need to ensure that the Web service can use either policy. What should you do?

- A. Create one policy file that contains a policy element, which contains a kerberosSecurity policy element and a mutualCertificateSecurity policy element.
- B. Create one policy file that contains a policy element for the kerberosSecurity policy and a policy element for the mutualCertificateSecurity policy.
- C. Create two policy files: one with the kerberosSecurity policy, and the other with the mutualCertificateSecurity policy.
- D. Create two policy files: one with the Extensions element, and the other with a single

policy element that contains both the kerberosSecurity policy element and the mutualCertificateSecurity policy element.

Answer: B

Explanation:

By declaring security policy in an XML file administrators are delegated the responsibility for specifying policy requirements. Two policy files are not required to implement both policies; therefore (C) and (D) are not the answer.

This file will contain a <policies> tag, and each security policy would have it's own <policy> tag; therefore (A) is incorrect, while (B) is the correct answer.)

QUESTION 6

You write an XML Web service. The XML Web service contains Web methods that return large amounts of non-sensitive public data. The data is transferred over the Internet. You need to be able to detect if the data was tampered with during transit. The implementation must be configurable at run time. Your solution must minimize the impact on the performance of the Web service. What should you do?

- A. Configure the virtual directory to require the Secure Sockets Layer (SSL) protocol.
- B. Apply Web Services Enhancements (WSE) 3.0 security to the Web service that is configured to use an X.509 certificate with the Sign-Only protection level..
- C. Apply Web Services Enhancements (WSE) 3.0 security to the Web service that is configured to use an X.509 certificate with the Sign and Encrypt protection level.
- D. Configure the virtual directory that hosts the XML Web service to use basic authentication.

Answer: B

Explanation:

This solution does not require the encryption of "non-sensitive public data," therefore the encryption of data using SSL (A), or using WSE 3.0 with Sign and Encrypt protection level (C) is not necessary.

Basic authentication (D) would require the signing in of clients without the need to encrypt that sign-on, but is not a part of the current solution.

Using WSE 3.0 with Sign-Only protection level (B) allows for the hashing, and digital signing of data, without the need to encrypt it. Thus meeting the needs for this solution.

QUESTION 7

You write a SOAP extension to monitor a deployed Web service. You need to deploy the SOAP extension to the Web service without requiring a change to the compiled assemblies. You cannot use reflection to deploy the SOAP extension. What should you do?

- A. Write a class that extends the SoapExtensionAttribute attribute, has the AttributeTargets.Method attribute applied, and returns an instance of the SOAP extension

in the ExtensionType property.

- B. Extend the SoapExtensionImporter class and reference the class by using the SoapExtensionTypeElement in the Web.config file.
- C. Extend the SoapExtension class. Override the Initialize method to add the SOAP extension to the Web service by using the SoapExtensionReflector class.
- D. Add the SOAP extension to the Web service's Web service description language (WSDL) by using the WsdlHelpGeneratorElement element in the Web.config file.

Answer: B

Explanation:

The solution states that the deployment must be made without a change to the compiled assemblies, therefore extending a class (A) and (C) does not meet this criterion, neither would adding a SOAP extension to the WSDL (D).

Only by using the web.config file to reference the class would the requirements be met.

QUESTION 8

You call a method in a Web service. The following exception is thrown in the Web service client. `System.Web.Services.Protocols.SoapException: Server was unable to process request. --> System.NullReferenceException: Object reference not set to an instance of an object.` You discover that it is the following line of code that throws the exception. `if (Session ["StoredValue"] == null).`

You need to ensure that the method runs without throwing the exception. What should you do?

- A. Add the following elements to the System.Web section of the Web.config file. `<httpModules><add name="Session" type="System.Web.SessionState.SessionStateModule" /></httpModules>`
- B. Modify the WebMethod attribute in the Web service so that the EnableSession property is set to True.
- C. In the client code for the Web service's proxy object, assign a new instance of the `System.Net.CookieContainer` object to the CookieContainer property.
- D. Add the following element to the System.Web section of the Web.config file. `<sessionState mode="InProc" />`

Answer: B

Explanation:

In order to store session state an XML Web Service must inherit from `WebService` and have a `WebMethodAttribute` applied to the Web Service Method, setting the `EnableSession` property to true (D). This property is set to false by default.

Not D: D do not set the EnableSession Property to true.

QUESTION 9

A Web service exposes a method named `GetChart` that returns an image. The data used to generate the image changes in one-minute intervals. You need to minimize the average

time per request for CPU processing. What should you do?

- A. Set the CacheDuration property on the WebMethod attribute of the GetChart method to 60.
- B. Set the BufferResponse property on the WebMethod attribute of the GetChart method to False.
- C. Set the CacheDuration property on the WebMethod attribute of the GetChart method to 1.
- D. Set the BufferResponse property on the WebMethod attribute of the GetChart method to True.

Answer: A

Explanation:

The BufferResponse property gets or sets whether the response for a given request is buffered, which would not minimize the average time per request, since it deals with a specific request, so answers (B) and (D) should not be considered.

The CacheDuration property gets or sets the number of seconds the response should be held in the cache, and since the image only changes in one-minute intervals, the best answer would be 60 seconds (A).

QUESTION 10

A Web service application provides security data about employees to applications that control access to company facilities. The Web service is accessed by using TCP and is protected by using Web Services Enhancements (WSE) 3.0 security for message encryption. The company has added fingerprint readers to grant employees access to the facilities. Images of employee fingerprints are maintained by the Web service application. You need to ensure that the existing WSE security encryption policy can be applied to the fingerprint image. Your solution must minimize the size of the Web service message. What should you do?

- A. Configure the Web service to use Message Transmission Optimization Mechanism (MTOM) to pass the binary fingerprint image.
- B. Configure the Web service to use base64 encoding to pass the binary fingerprint image.
- C. Create a SOAP filter to manage encryption for the message.
- D. Create a SOAP extension to manage encryption for the message.

Answer: A

Explanation:

When a programmer encodes binary data as XML, as in this problem, it yields a large, or 'fat' file that makes processing in XML Web Services almost unmanageable. The WSE 3.0 MTOM (Message Transmission Optimization Mechanism) uses XOP (XML-binary Optimized Packaging) to provide the binary data and the XML data as a package, thus letting SOAP bindings speed up data transmission. Thus (A) is correct.

QUESTION 11

A Windows Forms application calls in to a Web service named SensitiveData. The project has a Web reference named SensitiveDataWS. The code uses a class of type SensitiveDataWS.Service. SensitiveDataWS.Service is a proxy to the Web service. An administrator reports that users running the client application receive a SoapHeaderException exception with the following message text: "Security requirements are not satisfied because the security header is not present in the incoming message".

You discover that the Web Services Enhancements (WSE) 3.0 policy file for the Web service was changed to require the encryption of SOAP messages. You acquire the X.509 certificate that is used for encryption in the Web service. You need to ensure that the Windows Forms application meets the new security requirements of the Web service. What should you do?

- A. In each Web method, assign an instance of the X509Certificate class that is initialized with the acquired X.509 certificate, to the ClientCertificates property of the SensitiveDataWS.Service class.
- B. Apply WSE security to the Windows Forms project that is configured to use the acquired X.509 certificate with the Sign and Encrypt protection level. Renew the Web reference and modify the code to use the new SensitiveDataWS.ServiceWse class.
- C. Apply WSE security to the Windows Forms project that is configured to use the acquired X.509 certificate with the Sign-Only protection level. Renew the Web reference and modify the code to use the new SensitiveDataWS.ServiceWse class.
- D. In the Windows Forms project, create a custom class that inherits from the SoapHeader attribute class with a public property of type X509Certificate. Initialize the public property with the acquired X.509 certificate. Apply the created attribute to code in the client application that calls the Web service. Initialize the attribute with the name of the custom class.

Answer: B

Explanation:

The problem specifically states that WSE 3.0 will be used to encrypt SOAP messages. Only the Sign and Encrypt protection level (B) will accomplish this.

QUESTION 12

A SOAP message has the following body.

```
<soap:Body> <tns:Greeting>  
<Person href="#id1" /> </tns:Greeting> <tns:User id="id1" xsi:type="tns:User">  
<GivenName xsi:type="xsd:string">givenname</GivenName>  
<SurName xsi:type="xsd:string">surname</SurName>  
</tns:User></soap:Body>
```

You need to configure the Greeting method to accept the SOAP message. Which code segment should you use?

- A. <WebMethod()> _<SoapDocumentMethod(Use:=SoapBindingUse.Encoded)>
_Public Function Greeting(<XmlElement("Person")> ByVal user As User) ...
End Function
- B. <WebMethod()> _<SoapDocumentMethod(Use:=SoapBindingUse.Literal)>
_Public Function Greeting(<SoapElement("Person")> ByVal user As User) ...
End Function
- C. <WebMethod()> _<SoapDocumentMethod(Use:=SoapBindingUse.Literal)>
_Public Function Greeting(<XmlElement("Person")> ByVal user As User) ...
End Function
- D.<WebMethod()> _<SoapDocumentMethod(Use:=SoapBindingUse.Encoded)>
_Public Function Greeting(<SoapElement("Person")> ByVal user As User) ...
End Function

Answer: D

Explanation:

This question tests knowledge of the SoapDocumentMethodAttribute, specifically the Use property, and what type of element should be passed as a parameter to the method. The Use property determines whether parameters are documented as Encoded or Literal. Encoded refers to formatting parameters using SOAP encoding. Literal refers to formatting the parameters using a pre-defined XSD schema for each parameter, thus (B) and (C) would be wrong.

Since the problem requires sending a SOAP encoded message, the Use property must be Encoded, and the object passed as a parameter must be a SoapElement, thus (D) is correct.

QUESTION 13

You use Microsoft Visual Studio 2005 to create a custom Web service discovery system that contains Disco files for your Web services. You need to have the Disco file for each of your Web services for auditing purposes.

What are two possible ways to achieve this goal? (Each correct answer presents a complete solution. Choose two.)

- A. Append the Disco parameter to the Web service's .asmx URL's querystring and save the returned data.
- B. Open the Web service's Web services description language (WSDL) file and copy the method information into the Disco file for that Web service.
- C. Append the Web services description language (WSDL) parameter to the Web service's .asmx URL's querystring and save the returned data.
- D. Add a reference to the Web service in Visual Studio 2005 and use the generated Disco file.

Answer: A,D

Explanation:

Copying the WSDL files information does not constitute a proper DISCO document, so

(B) is incorrect.

Adding a reference to a Web Service does generate a proper DISCO document (D), and is correct.

So, which parameter when appended to the Web service's .asmx URL's query string will the Disco or WSDL parameter create a DISCO file? The WSDL parameter (C) creates a service description, while the Disco parameter creates a Discovery document, so (A) is also correct.

QUESTION 14

You write a Web service that is called by two client applications named Client1 and Client2. The Web service must return a SOAP header that contains the date and time that the method executed. You discover that the date and time information for Client1 must be stored in a different XML format from the format used for Client2. The client application that is making the request is specified as a property of the SOAP header. You need to ensure that date and time information is stored in the correct format for each client application. Your solution must not change the way that ASP.NET handles SOAP requests. What should you do?

- A. Change the EncodedRelay property to the name of the method that will create the correct XML formatting for each client application.
- B. Implement the IXmlSerializable interface in the SOAP header class to return the correct XML format for each client application in the WriteXml method.
- C. Add the SoapElementAttribute attribute with the correct element name for the client application to the field that contains the date and time information.
- D. Create two XmlSerializer classes and use the WriteHeaders method of the SoapHeaderHandling class to format the XML for each client application.

Answer: B

Explanation:

The IXmlSerializable.WriteXml method (B) writes out the XML representation of an object, which can include the way that the values within the object are formatted.

The requirements state that the way that ASP.NET handles SOAP requests may not change, therefore using the WriteHeaders method of the SoapHeaderHandling class (D) does not meet the requirements.

The EncodedRelay property (A) only applies to SOAP 1.2 and indicates whether the recipient of the message forwards this message to the next node downstream. The .NET Framework does not support SOAP intermediaries, so this property is ignored, and is incorrect.

The SoapElementAttribute (C) is applied to a public field to direct the XmlSerializer to serialize the field as an Encoded SOAP XML element, but provides no functionality to distinguish between the formatting of the field.

QUESTION 15

You create a serviced component. You install the component into the COM+ catalog. COM+ runs on the server. A Windows-based application that is installed on multiple

desktop computers must use the component. The component must run on the server, but the Windows-based application must send component method calls over the network to the component. The communications protocol used is DCOM. You need to ensure that the Windows-based application can connect to the component. What should you do?

- A. In the client application, create the serviced component using the Marshal class to bind to the following moniker."dcom://MyComponents.MyServicedComponent"
- B. Generate a proxy component for the serviced component and install it on the desktop computers that are running the client application. In the client application, create a reference to the serviced component's assembly.
- C. Generate a proxy component for the serviced component and install it on the desktop computers that are running the client application. In the client application, create a reference to the serviced component's type library.
- D. In the client application, create the serviced component using the Marshal class to bind to the following moniker."tcp://MyComponents.MyServicedComponent"

Answer: B

Explanation:

The Marshal class provides a collection of methods for allocating unmanaged memory, copying unmanaged memory blocks, and other interop services. The question makes no reference that the serviced component created runs as unmanaged code, therefore (A) and (D) are unlikely to be correct. Also, (D) would be binding to TCP not DCOM in accordance with the question, and is doubly incorrect.

Generating a proxy component is the most likely method for allowing Windows-based applications to send method calls to the Serviced Component, and since the question states that you created it, it is likely to be a managed serviced component, so the most likely answer is to set a reference to the serviced component's assembly (B), not to its type library (C).

QUESTION 16

You create a .NET Framework remoting application that provides stock information to customers. The server component raises an event on the client computer when certain conditions are met. You need to ensure the server raises exactly one event for each client application that is registered for the event. What should you do?

- A. Configure the server class as a Singleton Server Activated Object (SAO) and check for duplicate client delegate methods before raising the event.
- B. Configure the server class as a SingleCall Server Activated Object (SAO) and check for duplicate client delegate methods before raising the event.
- C. Configure the server class as a Client Activated Object (CAO) and check for duplicate client delegate methods before raising the event.
- D. Configure the server class as a Client Activated Object (CAO) and override the CreateObjRef method to check for duplicate client delegate methods before raising the event.

Answer: A

Explanation:

Singleton Server Activated Objects (A) always only have one instance, regardless of how many clients there are for the object, making this the ideal means of solving this problem. A Client Activated Object is created on the server when the client calls new or Activator.CreateInstance. In this scenario we want to ensure that the server raises exactly one event for each client application. This would indicate a need to better control the number of instances created, so neither (C) nor (D) would be correct.

A Server Activated Object is created by the server only when they are needed, when the client invokes the first method on the proxy. SingleCall objects (B) have one instance of the class for each client invocation method, making it more difficult to ensure exactly one event for each client application.

QUESTION 17

You create a Web service. The Web service must be deployed on a remote Web server. You need to ensure that the deployment does not place the source code for the Web service on the Web server. What should you do?

- A. Use the ASP.NET Compilation tool (Aspnet_compiler.exe) to compile the Web service locally. Move the resulting files to the Web server.
- B. Add a CodeBehind attribute to the @WebService directive. Rebuild the Web service project and deploy it by using the Copy Web Site Wizard.
- C. Add a Class attribute to the @WebService directive. Rebuild the Web service project and deploy it by using the Copy Web Site Wizard.
- D. Move the contents of the development Web site to the Web server.

Answer: A

Explanation:

The Aspnet_compiler.exe (A) can compile an ASP.NET Web application, either in place or remotely, and can be performed in two different ways; one that removes all source files, and one that retains markup files.

The Copy Web Site Tool (B) and (C) copies files between a source site and a remote site, but does not have a facility to not copy source code files, unless you copy files individually. Neither would be the best solution.

Moving the contents (D) does not include information regarding which contents will be moved, and would not enable the web service to operate, if the source code files were not also copied.

QUESTION 18

You write a client application that uses a Web service. The Web service is hosted in a Windows service application. The Web service can be called only by using TCP. You are provided with the proxy that is enabled by Web Services Enhancements (WSE) 3.0. The proxy was generated by adding a Web reference by using Microsoft Visual Studio 2005. You need to ensure that the proxy calls the Web service correctly. Which two actions

should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Set the Action object in the SOAP request's addressing headers to the name of the Web service method you want to call.
- B. Set the Action object in the SOAP request's addressing headers to the TCP address of the Web service you want to call.
- C. On the Destination property of the proxy, set the TransportAddress property to the TCP address of the hosted Web service.
- D. Change the URL of the proxy to the TCP address of the hosted Web service.

Answer: A,D

Explanation:

The Action object in the SOAP request's addressing headers must be set to the Web service method to be called, hence (A) is correct, while (B) is not.

When generating a WSE proxy from Visual Studio 2005 by using the 'Add Web Reference' (as long as WSE is enabled in the client application - as inferred by the wording of the question) you simply change the address of the proxy to the TCP address using either the Url or destination property, therefore (C) is incorrect, and (D) is correct.

QUESTION 19

An administrator reports that when a client application runs, the application throws a RemotingException exception with the message:

"Object '/40b4b673_e739_43df_abe4_ee269ff67173/0t_g9yvtvi_lgue2i9q5qrni_1.rem' has been disconnected or does not exist at the server". You discover the following information: The object causing the exception is configured as a Client Activated Object (CAO). The exception is thrown only if the client application is idle for more than five minutes. If the client application is idle for nine minutes, the client application shuts down and the user is logged out. You need to ensure that the CAO is available to the client application. What should you do?

- A. Add the following XML to the App.config file for the client application.
`<configuration> <system.runtime.remoting>
<application>
<lifetime leaseTime="999999999999" />
</application> </system.runtime.remoting></configuration>`
- B. Add the following code to the CAO class.
`public override object InitializeLifetimeService() { ILease lease = (ILease)base.InitializeLifetimeService();
if (LeaseState.Initial == lease.CurrentState) {
lease.RenewOnCallTime = TimeSpan.FromMinutes(10); }
return lease;
}`
- C. Add the following code to the CAO class.
`public override object InitializeLifetimeService() { ILease lease = (ILease)base.InitializeLifetimeService();
if (LeaseState.Initial == lease.CurrentState) {`

```
lease.SponsorshipTimeout = TimeSpan.FromMinutes(10); }  
return lease;  
}
```

D. Add the following XML to the App.config file for the client

```
application.<system.runtime.remoting> <application>  
<lifetime leaseManagerPollTime="10m" /> </application></system.runtime.remoting>
```

Answer: B

Explanation:

The leaseTime attribute (A) specifies the lease time for client-activated objects serviced by the application, while the leaseManagerPollTime (D) specifies amount of time the lease manager sleeps after checking for expired leases, neither of which would affect a specific CAO.

The InitializeLifetimeService method obtains a lifetime service object to control the lifetime policy of an instance. On an ILease object the SponsorshipTimeout property (C) gets or sets the amount of time to wait for a sponsor to return with a lease renewal time, which doesn't meet the requirements of the question.

The RenewOnCallTime property (B) of an ILease object, however, gets or sets the amount of time by which a call to the remote object increases the CurrentLeaseTime, and is what is required for this requirement.

QUESTION 20

You create an assembly named SimpleMathLib.dll. SimpleMathLib.dll contains a namespace named Simple Math. Simple Math contains a .NET Framework remoting class named SimpleMathClass that inherits from the MarshalByRefObject class. You deploy the SimpleMathLib assembly to the Bin folder under an IIS server virtual directory named SimpleMathHost that is on a server named Server1. You need to configure the server application to initialize the SimpleMathClass class so that the class can be called by remoting client applications. What should you do?

A. Add the following XML code to the configuration element of the Web.config file in the SimpleMathHost virtual directory.

```
<configSections> <section name="simpleMath"  
type="SimpleMath.SimpleMathClass,SimpleMathLib" /></configSections><simpleMath  
mode="SingleCall"> <url>http:  
//server1/SimpleMathHost/SimpleMath.soap</url></simpleMath>
```

B. Add the following XML to the configuration element of the Web.config file in the SimpleMathHost virtual directory.

```
<system.runtime.remoting> <application>  
<service>  
<wellknown mode="SingleCall"  
type="SimpleMath.SimpleMathClass, SimpleMathLib"  
objectUri="SimpleMath.soap"/>  
</service> </application></system.runtime.remoting>
```

C. Add the following XML to the configuration element of the Web.config file in the

SimpleMathHost virtual directory.

```
<appSettings> <add key="SimpleMath.SimpleMathClass,SimpleMathLib"
value="http: //server1/SimpleMathHost/SimpleMath.soap" /></appSettings>
```

D.Add the following XML code to the configuration element of the Web.config file in the SimpleMathHost virtual directory.

```
<system.runtime.remoting> <application>
<client>
<wellknown
type="SimpleMath.SimpleMathClass, SimpleMathLib"
url="http: //server1/SimpleMathHost/SimpleMath.soap"/>
</client> </application></system.runtime.remoting>
```

Answer: B

Explanation:

Neither adding a <configSections> element (A), nor adding an <appSettings> element (C) can be used to configure the server application for remoting, so neither of these answers is correct.

To put custom settings in remoting application configuration files requires a <system.runtime.remoting> section. Since the requirements of the question are for the server application to initialize the class, answer (D) does not meet this requirement, because it has the client initialize, while (B) creates a Server Activated object, and is correct.

QUESTION 21

You are writing a .NET Framework remoting client application. The server application raises events to the client application using an interface named IBroadcaster and an event wrapper. The event wrapper and the interface are located in an assembly named General that is common to the client and server. The server configuration file is as follows:

```
<system.runtime.remoting>
<application>
<channels>
<channel ref="http" port="2020">
<serverProviders>
<formatter ref="binary" typeFilterLevel="Full" />
</serverProviders>
<clientProviders>
<formatter ref="binary" />
</clientProviders>
</channel>
</channels>
<service>
<wellknown mode="Singleton" type="Server.Broadcaster,
Server" objectUri="Broadcaster.soap" />
</service>
</application>
```


</system.runtime.remoting>

You need to ensure that the server can raise events on the client application. What should you do?

- A. Configure the remoting configuration file for the client application to use Server.Broadcaster, Server as the type in the wellknown element. Specify 0 as the port for the channel.
- B. Configure the remoting configuration file for the client application to use Server.Broadcaster, Server as the type in the wellknown element. Do not specify a port for the channel.
- C. Configure the remoting configuration file for the client application to use General.IBroadcaster, General as the type in the wellknown element. Do not specify a port for the channel.
- D. Configure the remoting configuration file for the client application to use General.IBroadcaster, General as the type in the wellknown element. Specify 0 as the port for the channel.

Answer: D

Explanation:

The <wellknown> element for the client contains information about server-activated (wellknown) objects the application wants to consume. When a client consumes a <wellknown> element, the parent <client> element does not need a url attribute, as in this question. The client <wellknown> element specifies the full name of the object and the assembly name of the type implementation on the client, therefore (A) and (B), since they use Server.Broadcaster would be incorrect.

The configuration of the client also must specify a port number, so (D) would be the correct answer.

QUESTION 22

You write a Web service that processes multiple SOAP headers, including one named Header

- A. You examine the HTTP log and find that an unknown SOAP header is included in the client application requests. Your Web service must process all headers. You need to indicate to the client application that the unknown SOAP header is unexpected. What should you do?
- A. Check the MustUnderstand property of the unknown SOAP header. If the property value is set to True, throw a SoapHeaderException exception.
 - B. Set the DidUnderstand property of the unknown SOAP header to False.
 - C. Check the DidUnderstand property of the unknown SOAP header. If the property value is set to True, throw a SoapHeaderException exception.
 - D. Set the MustUnderstand property of the unknown SOAP header to False.

Answer: B

Explanation:

The SoapHeader.MustUnderstand property gets or sets a value indicating whether the SoapHeader must be understood, which is not what is being asked for, therefore (A) and (D) are incorrect.

The SoapHeader.DidUnderstand property gets or sets a value indicating whether an XML Web Service method properly processed a SOAP Header. If you set the DidUnderstand property to false, and if it remains false after the method returns, a SoapHeaderException is thrown back to the XML Web service, therefore (B) is correct.

QUESTION 23

A message queue named SecureQueue requires each incoming message to be encrypted. You need to ensure that a message can be sent to the SecureQueue queue without an exception being thrown. What should you do?

- A. Set the EncryptionAlgorithm property on the message to a value from the EncryptionAlgorithm enumeration.
- B. Set the UseEncryption property on the message to True.
- C. Set the HashAlgorithm property on the message to a value from the HashAlgorithm enumeration.
- D. Use the CryptoStream class to encrypt the Body property on the message.

Answer: B

Explanation:

The EncryptionAlgorithm property of the Message class (A) gets or sets the algorithm used to encrypt the body of a private message, but does not require the encryption to occur before the message is sent.

The UseEncryption property of the Message class (B) gets or sets a value indicating whether to make the message private, by setting this value to true will ensure that all messages sent to SecureQueue are encrypted, and is the correct answer.

The HashAlgorithm property of the Message class (C) gets or sets the algorithm used when authenticating or creating a digital signature for a message, which has nothing to do with encryption, nor does it require the message to be hashed.

The CryptoStream class (D) is not directly used with the Message class.

QUESTION 24

A Web service returns a Node object X that references Node object Y. Node object Y also references Node object X. When the Web service method is called in a way that returns Node object X, the following exception is thrown.

System.InvalidOperationException: A circular reference was detected while serializing an object of type Node. You need to ensure that the Web service method runs without generating the exception. Your code must preserve the circular reference between the Node objects. What should you do?

- A. Add the following attribute to the Web service method: <SoapRpcMethod(Binding:="Encoded")>

- B. Add the following attribute to the Web service method.<SoapRpcMethod(Binding:="Literal")>
C. Add the following attribute to the Web service method.<SoapRpcMethod(Use:=SoapBindingUse.Literal)>
D. Add the following attribute to the Web service method.<SoapRpcMethod(Use:=SoapBindingUse.Encoded)>

Answer: D

Explanation:

The SoapRpcMethod attribute can be applied to individual XML Web Service methods to indicate the binding operation implemented by a particular XML Web Service method, which has nothing to do with the requirements of this question, therefore (A) and (B) are incorrect.

The SoapDocumentMethod attribute specify that SOAP messages to and from the method use Document formatting. The Use property determines whether parameters are formatted Encoded (D) or Literal (C). In this case, we need to preserve the circular reference between the Node objects, needing to use Encoded formatting.

NOTE: Uncertainty. The SoapRpcMethod attribute does not have a Use attribute, although the SoapDocumentMethod attribute does.

QUESTION 25

You are converting an application to use .NET Framework remoting. The server portion of the application monitors stock prices and contains a class named StockPriceServer, which is a Server Activated Object (SAO). The client computer interacts with the server using a common assembly. When the server attempts to raise an event on the client computer, the server throws the following exception.

System.IO.FileNotFoundException.

You discover that the event delegate is not being called on the client computer. You need to ensure that the server application can raise the event on the client computer. What should you do?

- A. Add the event delegate to the common assembly. Implement the Add delegate and the Remove delegate methods of the event in the StockPriceServer class to reference the delegate method in the client application.
B. Raise the event using the BeginInvoke method and pass a reference to the client computer.
C. In the common assembly, add an interface that contains the event and a method to raise the event. Implement that interface in the StockPriceServer class and use the interface's event to register the delegate message on the client computer.
D. Add the Serializable attribute to the StockPriceServer class and change the event to use one of the standard common language runtime (CLR) delegates.

Answer: C

Explanation:

To cause an event to be received on the server, an event has to be both declared and raised on the client. The only answer that takes both of these actions into account is (C), and is therefore correct.

QUESTION 26

You are writing a Web service application that uses Web Services Enhancements (WSE) 3.0 policies. The Web service request and response data must be signed. Routing occurs between the client and server computers, and uses the Action SOAP header of the SOAP messages. You need to ensure that the SOAP signature is not invalidated when the SOAP message is routed. What should you do?

- A. Sign the SOAP request with a Username token that includes a password to allow the SOAP message to be signed again during routing.
- B. In the policy file, set the signatureOptions attribute to IncludeSoapBody.
- C. Sign a SOAP request that encapsulates the initial SOAP request.
- D. In the policy file, set the signatureOptions attribute to IncludeAddressing.

Answer: B

Explanation:

Since routing occurs between the client and server computers the question does not infer that re-signing can be done during routing, so (A) and (C) is not correct.

If in the policy file, the signatureOptions is set to IncludeAddressing (D) it will only sign the <Action>, <FaultTo>, <From>, <MessageId>, <Recipient>, <RelatesTo>, <ReplyTo> and <To> headers, but the question asks that the signature for the message not be invalidated, so (D) is not correct.

The signatureOption of IncludeSoapBody (B) indicates that the SOAP body should be signed, which fulfills the requirement.

QUESTION 27

You extend the SoapExtension class to implement a SOAP extension. You need to override the ChainStream method to implement the SOAP extension functionality correctly. What should you do?

- A. Register event listeners that will modify data for the input and output streams that the SOAP extension will use.
- B. Create and return a new stream that will be used to send data down the chain. Save a reference to the stream that is coming from higher on the chain.
- C. Save references to the input and output streams that the SOAP extension will use to send and receive data.
- D. Add the SOAP extension to the position in the SOAP extension chain that assigns the appropriate priority.

Answer: B

Explanation:

The Chain Stream method returns a Stream representing a new memory buffer that the SOAP extension can modify, so only answer (B) makes sense in this context, and is correct.

QUESTION 28

You create a Web service that will be deployed to a production Web server. You need to ensure that the first Web service request returns a response in the shortest amount of time possible. What should you do?

- A. Run the ASP.NET Compilation tool (Aspnet_compiler.exe) on the Web service project. Copy the generated files to the Web server.
- B. Build the Web service project on the development platform. Deploy the .asmx files and the contents of all subdirectories to the Web server.
- C. Set the Code Behind attribute on the @WebService directive. Deploy the .asmx files and the source code files to the Web server.
- D. In the Web.config file, set the batch attribute in the compilation element to True. Deploy the .asmx files and the source code files to the Web server.

Answer: A

Explanation:

Neither building the web service (B) nor deploying with source code files (C) precompiles the web service functionality for immediate execution in the shortest time. Batch compilation (D) is still available in ASP.NET 2.0 and removes the delay on the first page request, but creates a longer cycle time on startup. So, the first request is going to take significantly longer, but pages will thereafter be immediately available. The ASP.NET Compilation tool (A) enables a complete compilation of the entire application, helping performance because end-users do not encounter a delay on the first request to the application while the application is compiled, which is what is required in this case.

QUESTION 29

You are creating a Windows-based application that allows users to store photographs remotely by using a Web service. The Web service is configured to use Web Services Enhancements (WSE) 3.0, and it has the following configuration XML in its Web.config file.

```
<microsoft.web.services3> <messaging>  
<mtom serverMode="always"/>  
</messaging></microsoft.web.services3>
```

You need to configure the application to ensure that a photograph can be sent to the Web service.

What should you do?

- A. Add the following XML to the application's App.config file.

```
<microsoft.web.services3> <messaging>  
<mtom clientMode="Off" /> </messaging></microsoft.web.services3>
```

- B. Add the following XML to the application's App.config file.
`<microsoft.web.services3> <messaging>
<mtom serverMode="always" /> </messaging></microsoft.web.services3>`
- C. Add the following XML to the application's App.config file.
`<microsoft.web.services3> <messaging>
<mtom clientMode="On" /> </messaging></microsoft.web.services3>`
- D. Add the following XML to the application's App.config file.
`<microsoft.web.services3> <messaging>
<mtom serverMode="never" /> </messaging></microsoft.web.services3>`

Answer: C

Explanation:

This question tests whether or not to use MTOM (Message Transmission Optimization Mechanism), and where to do so. One of the enhancements of WSE 3.0 is the addition of MTOM as a technology to efficiently transmit binary data (like photos) with SOAP messages. Since this is what is required in this scenario, neither (A) nor (D) could be correct.

The clientMode attribute specifies that messages are either MTOM encoded or not, unless client code explicitly sets the RequireMtom property of the proxy class to the opposite. In this scenario this make the most sense.

The serverMode attribute specifies whether all incoming and outgoing SOAP messages must be MTOM encoded. If a SOAP request is received that is not MTOM encoded, an HTTP error 415: "Media unsupported" is returned to the sender. Since the question indicates that this has already been set at the Web service it is not the correct answer.

QUESTION 30

An administrator reports that a .NET Framework remoting application named MyServer has poor performance. You want to observe the application's performance during a high-activity time period. You run System Monitor on the computer that is running MyServer. You need to add the correct performance counter to the details view of System Monitor. What should you do?

- A. Add the MyServer instance of the Remote Calls/sec counter from the .NET CLR Remoting performance object.
- B. Add the _Global_ instance of the Total Remote Calls counter from the .NET CLR Remoting performance object.
- C. Add the MyServer instance of the Total Remote Calls counter from the .NET CLR Remoting performance object.
- D. Add the _Global_ instance of the Remote Calls/sec counter from the .NET CLR Remoting performance object.

Answer: A

Explanation:

The scenario identifies that we need to check the poor performance of MyServer, so

accessing the Global_instance object would not work for the scenario, so (B) and (D) are incorrect.

The Remote Calls/sec counter (A) displays the number of remote procedure calls invoked per second, which focuses on the performance of the application, and makes this answer the most correct.

The Total Remote Calls counter (C) displays the total number of remote procedure calls invoked since the application started, which does not look at the specific performance of the application, and makes this answer wrong.

QUESTION 31

A client application must connect to a .NET Framework remoting class that is running in a Windows service that is deployed to a computer named APPSERVER1. The class is named SimpleMathClass and it exists within a namespace named SimpleMath in an assembly named SimpleMathLib. It is exposed through the TCP and it is configured with the objectUri attribute set to SimpleMath.rem. You write the following XML configuration settings in the App.config file of the client application.

```
<configuration> <system.runtime.remoting>
<application>
</application>
</system.runtime.remoting></configuration>
```

You need to ensure that the client application can connect to the remoting class. What should you do?

A. Add the following XML configuration settings to the application element of the client application's App.config file. <service name="SimpleMathHost"> <wellknown mode="Singleton"

type="SimpleMath.SimpleMathClass, SimpleMathLib"
objectUri="SimpleMath.rem"/></service>

B. Add the following XML configuration settings to the application element of the client application's App.config file. <service name="SimpleMathHost"> <wellknown mode="SingleCall"

type="SimpleMath.SimpleMathClass, SimpleMathLib"
objectUri="SimpleMath.rem"/></service>

C. Add the following XML configuration settings to the application element of the client application's App.config file. <client> <wellknown type="SimpleMath.SimpleMathClass, SimpleMathLib"

url="tcp://APPSERVER1:3085/SimpleMath.rem"/></client>

D. Add the following XML configuration settings to the application element of the client application's App.config file. <client> <wellknown type="SimpleMath.SimpleMath, SimpleMath"

url="http: //APPSERVER1:3085/SimpleMathHost/SimpleMath.rem"/></client>

Answer: C

Explanation:

Client applications that connect to remote classes are set up in the App.config file in a

<client> element, so neither answer (A), nor answer (B) are correct answers.

Answer (D) is incorrect on two counts; first, the parameters for the type attribute of the wellknown element require the namespace.class name followed by the assembly name, and secondly, the scenario specified the tcp protocol, while this answer uses http.

Answer (C) is correct because it correctly identifies the appropriate type, and the correct url for the class.

QUESTION 32

You are debugging a Web service application that uses Web Services Enhancements (WSE) 3.0. When you attempt to view tracing information for the Web service requests, you discover that the log file is not being updated as expected. You need to ensure that the log file is updated. What should you do?

A. Add the following XML to the configuration section of the Web.config file.

```
<system.diagnostics> <switches>  
<add name="WseTracing" value="InputTrace.webinfo"/>  
</switches></system.diagnostics>
```

B. Add the following XML to the configuration section of the Web.config file.

```
<microsoft.web.services3> <trace enabled="true" input="InputTrace.webinfo"  
/></microsoft.web.services3>
```

C. Add the following XML to the system.web section of the Web.config file.

```
<trace enabled="true" configSource="InputTrace.webinfo"  
writeToDiagnosticsTrace="true"/>
```

D. Add the following XML to the configuration section of the Web.config file.

```
<microsoft.web.services3> <diagnostics>  
<trace enabled="true" input="InputTrace.webinfo" />  
</diagnostics></microsoft.web.services3>
```

Answer: D

Explanation:

First of all, there is a trace element in the system.web section of the web.config file (C), however, neither the configSource, nor the writeToDiagnosticsTrace attributes are part of the default schema, so this answer is false.

The <add> element of the <switches> element of the <system.diagnostics> element (A) is used to change the level of a trace switch, so this answer is incorrect.

Since the scenario identifies that the technology being used is WSE 3.0, the <trace> element is required to be inside of a <diagnostics> element, therefore (B) is incorrect, while (D) is correct.

QUESTION 33

You are creating a Web service by using ASP.NET. You need to ensure that the Web Services Description Language (WSDL) file that is generated is Web services interoperability (WS-I) compliant and that it allows message validation. What should you do?

- A. Add the following attribute definition to the Web service.
<SoapRpcMethod(Use:=SoapBindingUse.Literal)>
- B. Add the following attribute definition to the Web service.
<SoapDocumentMethod(Use:=SoapBindingUse.Literal)>
- C. Add the following attribute definition to the Web service.
<SoapDocumentMethod(Use:=SoapBindingUse.Encoded)>
- D. Add the following attribute definition to the Web service.
<SoapRpcMethod(Use:=SoapBindingUse.Encoded)>

Answer: B

Explanation:

In this scenario we can eliminate the SoapRpcMethod attribute answers (A and D) for two reasons; first, the attribute does not provide for a Use property, and secondly, the WS-I Basic Profile is an attempt to limit the flexibility of the Web service specifications in exchange for greater interoperability. This means that RPC encoded messages are not allowed.

Because of the emphasis on interoperability, your web service should subscribe to industry standards (like XSD) which is supported by the Literal SoapBindingUse (B), but not by the Encoded SoapBindingUse (C). Thus, answer B is correct.

QUESTION 34

You build Web services by using Microsoft Visual Studio 2005. Some of the Web services are hosted in IIS. You need to ensure that remote systems cannot dynamically discover which Web services are hosted in IIS. What should you do?

- A. In the Application Mappings section of the IIS Application Configuration dialog box, remove the .asmx extension.
- B. In the Application Mappings section of the IIS Application Configuration dialog box, remove the .vsdisco extension.
- C. In the Application Mappings section of the IIS Application Configuration dialog box, remove the .axd extension.
- D. In the Application Mappings section of the IIS Application Configuration dialog box, remove the .ashx extension.

Answer: B

Explanation:

Since the scenario requires no dynamic discovery, only the .vsdisco extension enables this, and so answer (B) is the correct answer.

Removing any of the other extensions would disable the ability of IIS to run the Web services application.

QUESTION 35

You create a Web service. The method in the Web service maintains session information between calls. When a client invokes the method, the following exception is

thrown.System.Web.Services.Protocols.SoapException: Server was unable to process request. ---> System.NullReferenceException: Object reference not set to an instance of an object. You need to ensure that the Web service method can be called without generating an exception.

What should you do?

- A. Set the Conforms To property in the WebServiceBindingAttribute attribute to WsiProvfiles.BasicProfile1_1.
- B. Set the Enable Session property of the Web Method attribute to True.
- C. Use the WebService.Session object instead of the HttpContext.Session object to access the session variables.
- D. Set the AllowAutoRedirect property on the proxy class on the Web service client to True.

Answer: B

Explanation:

Sessions are not enabled for Web services by default. To enable maintaining session information between calls, set the EnableSession property of the WebMethod attribute (B) to true.

QUESTION 36

Users report that a Windows-based application does not run properly. When users attempt to complete a particular action, the following error message text appears.

Unable to find assembly 'myservices, Version=1.0.0.0,Culture=neutral, PublicKeyToken=29b5ad26c9de9b95'.

You discover that the error occurs when the Windows-based application attempts to call functionality in a serviced component that was registered by using the following command.
regsvcs.exe myservices.dll
You need to ensure that the application can call the functionality in the serviced component without throwing the exception. What should you do?

- A. Copy the serviced component assembly into the following folder.C:\WINDOWS\system32\Com
- B. Run the following tool from the command line.gacutil.exe /i myservices.dll
- C. Run the following tool from the command line.regasm.exe myservices.dll
- D. Copy the serviced component assembly into the following folder.C:\Program Files\ComPlus Applications

Answer: C

Explanation:

A serviced component is hosted by a COM+ application and must be accessible to that application. For accessibility, serviced components have the following registration and configuration requirements:

They must be strong-named.(RegSves would not work without this).

They must be registered in the Windows registry.

Type library definitions must be registered and installed into a specific COM+

Services added programmatically must be configured in the COM+catalog.(RegSvcs dose this).

Hence, the only requirement that has not been accounted for by having already run regsvcs, is that the assembly must be registered in the Windows registry, which is done with the Regasm.exe utility (C).

QUESTION 37

An application logs information about messages that pass through a message queue, including message contents. The application must run once for each message. You need to ensure that the application is executed for every message. Your solution must not prevent other processes from receiving the messages on the same queue. What should you do?

A. Create a Windows service application. Use the Peek method to retrieve message contents as messages arrive in the queue. Pass the message contents to the logging application.

B. Create a Message Queuing trigger that executes the logging application. Define the trigger rule to pass the message contents as a string parameter to the logging application.

C. Create a Message Queuing trigger that executes the logging application. Define the trigger rule to pass the message ID as a parameter to the logging application. Use the ReceiveById method on the MessageQueue object to retrieve the message content within the logging application.

D. Create a Windows service application. Use the Peek method to find out when a message arrives in the message queue. Use the ReceiveById method to retrieve the message contents. Pass the message contents to the logging application.

Answer: B

Explanation:

The Peek method of the Queue class is used to return a copy of the first message in a queue, without removing the message from the queue. Answer (A) says to call this method as each message arrives in the queue. Except for the first message received there is no guarantee that when a message arrives it will be in the first position in the queue, so this would not work.

For the same reason answer (D) would also not work - Peek is not designed to indicate when a message arrives in the queue.

This means that the best method for responding to a message as it comes into the queue is by creating a Message Queuing trigger that will execute the logging application. The MessageQueue classes ReceiveById method (C), however, receives the message that matches the given identifier, and removes it from the queue, contrary to the requirements of this solution. This makes passing the messages contents as a string parameter to the logging application (B) the only feasible answer.

QUESTION 38

A .NET Framework remoting application must host two linked-list collections, including

one collection of integers and one collection of strings. Both collections must be instances of a single collection type. The application must be configured so that there is only one instance of each collection at run time. You need to implement a linked-list collection type that meets these requirements.

What should you do?

- A. Create a generic type named `LinkedList` that inherits from the `ServiceComponent` class, implement linked-list behavior in the code, and then configure two instances of the generic in `SingleCall` mode in a remoting server.
- B. Create a generic type named `LinkedList` that inherits from `MarshalByRefObject` class, implement linked-list behavior in the code, and then configure two instances of the generic in `Singleton` mode in a remoting server host.
- C. Create two instance variables of type `System.Collections.Generic.LinkedList`. Host the two instance variables in a Web method for a Web service.
- D. Create a class named `LinkedList`, implement linked-list behavior in the code, and host two instances of the class in a Windows service.

Answer: B

Explanation:

Since this scenario expects strongly typed linked-lists, a generic type would be the most plausible solution, and implementing a custom class (D) would include the reproduction of work already done and available in the Framework.

The scenario specifically requires only one instance of each collection at run time, so answer (C) does not meet this requirement.

The option of creating a generic type is the most efficient, and by creating the instances in `Singleton` mode (B) you ensure that there will only be one instance of the two types at any one time, whereas creating them in `SingleCall` mode (A) would not guarantee this requirement being met.

QUESTION 39

A Web service exposes the following Web method. `<WebMethod(CacheDuration:=60)> _Public Function GetImage(ByVal imageId As String) As Byte() ...End Function` The Web method generates responses that are greater than 1 MB in size. You need to configure the Web method to minimize memory usage on the server. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Set the value of the `CacheDuration` property of the `WebMethod` attribute to 300.
- B. Apply a `SoapDocumentMethod` attribute to the method declaration with the `ParameterStyle` property set to `SoapParameterStyle.Bare`.
- C. Set the value of the `CacheDuration` property of the `WebMethod` attribute to 0.
- D. Apply a `SoapDocumentMethod` attribute to the method declaration with the `ParameterStyle` property set to `SoapParameterStyle.Wrapped`.
- E. Set the `BufferResponse` property of the `WebMethod` attribute to `False`.
- F. Set the `BufferResponse` property of the `WebMethod` attribute to `True`.

Answer: C,E

Explanation:

The CacheDuration property of the WebMethod attribute gets or sets the number of seconds the response should be held in the cache. Setting this to 300 (A) would cache the response on the server for five minutes, contrary to the directive to minimize memory usage on the server. Therefore setting this to 0 (C) would meet this requirement, and is correct.

The BufferResponse property of the WebMethod attribute gets or sets whether the response for this request is buffered. Setting this to true (F) would ensure that the response would be held until it is complete, or the buffer is full, and is contrary to the directive to minimize memory usage on the server. Therefore, setting it to false (E) ensures that the response is sent back to the client as it is serialized, and is correct.

Applying the SoapDocumentMethod attribute to a method specifies that SOAP messages to and from the method use Document formatting, and the ParameterStyle property of this attribute gets or sets whether parameters are encapsulated within a single XML element beneath the Body element in the SML portion of a SOAP message, and have nothing to do with minimizing memory usage on the server, and are therefore incorrect, whether using Bare (B), or using Wrapped (D).

QUESTION 40

You are debugging a .NET Framework remoting application in Microsoft Visual Studio 2005. Your solution contains three projects as follows: SimpleMathLib is a class library project. SimpleMathLib contains the remoting classes that will compile to a common assembly. SimpleMathHost is a Windows service project. SimpleMathHost is installed on your computer and it hosts SimpleMathLib. SimpleMathClient is a Windows Forms application project. SimpleMathClient is the remoting client application. When you run the SimpleMathClient application, you receive an error message. You set a breakpoint in the SimpleMathLib class library where you suspect the error. When you run SimpleMathClient in the debugger, the error occurs again, and the debugger does not stop on the breakpoint. You confirm that the breakpoint has been configured correctly. You need to ensure that you can hit the breakpoint and step through the code. What should you do?

- A. Attach to the w3wp.exe process.
- B. Copy the SimpleMathLib.pdb file from the SimpleMathLib project folder to the folder in which SimpleMathHost.exe is installed.
- C. Attach to the SimpleMathHost.exe process.
- D. Copy the SimpleMathLib.dll file to the global assembly cache and copy the SimpleMathLib.pdb file to the root folder on the system drive.

Answer: C

Explanation:

The W3WP.exe process (A) is the ASP.NET process, and since none of the projects are Web Application, Web Parts, or Web Services applications, this would not be an

appropriate option.

The Visual Studio debugger uses the path to the .pdb (project database) file in the EXE or DLL file to find the project.pdb file that holds debugging and project state information that allows incremental linking of a debug configuration of the program. Neither of the options for relocating the .pdb file make any sense in this scenario.

Since the SimpleMathHost is installed on your computer and it hosts the SimpleMathClient, which is generating the error, it makes sense to attach to the SimpleMathHost.exe process (C).

QUESTION 41

You are writing a Web service that will serve multiple client applications. Some client applications will send billing information as SOAP headers, but you do not know the format of these headers. You need to handle these SOAP headers so that the billing information is saved for analysis. What should you do? (Each correct answer presents part of the solution. Choose two.)

- A. Extend the WebService class and override the SoapVersion property to include the unknown headers.
- B. Add a variable that is an array of the SoapUnknownHeader header to the Web service class.
- C. Add the SoapHeaderAttribute attribute to the methods that are called by the client application with the WebMethodAttribute attribute applied.
- D. Extend the SoapUnknownHeader class and add the extended class as a SOAP header to the WebService class.

Answer: B,C

QUESTION 42

An application calls a Web method asynchronously by using the following code. (Line numbers are included for reference only.)

```
01
void ProcessData() {
02
    ProcessingService serviceProxy = new ProcessingService();
03
    IAsyncResult asyncResult = null;
04
    asyncResult = serviceProxy.BeginProcess(data, null, null);
05
    while (!asyncResult.IsCompleted) {
06
        Thread.Sleep(1000);
    }
08
09
    serviceProxy.EndProcess(asyncResult);
```


10 }

You need to ensure that the application can process and log any exceptions raised by the Web method. What should you do?

- A. Replace line 09 with the following code.

```
try {
serviceProxy.EndProcess(asyncResult);} catch (Exception ex) { LogException(ex);}
```
- B. Replace line 08 with the following code.

```
if (asyncResult.AsyncState is Exception)
LogException (asyncResult.AsyncState);
```
- C. Replace lines 05, 06, and 07 with the following code.

```
try { while
(!asyncResult.IsCompleted) {
Thread.Sleep(1000); }} catch (Exception ex) { LogException (ex);}
```
- D. Replace line 04 with the following code.

```
try { asyncResult =
serviceProxy.BeginProcess(data, null, null);} catch (Exception ex) { LogException (ex);}
```

Answer: A

Explanation:

The results of the Web Method will only be known after calling the EndProcess method, so trying to catch exceptions during processing (C) or at the beginning of processing (D) won't work.

The AsyncResult.AsyncState property (B) gets a user-defined object that contains information about the asynchronous operation. While it would be possible to create an Exception object that would be passed as the result of this, it is not written in the question, and would be at best a very round about way to accomplish the result.

The most efficient way to determine if a Web Method results in an Exception is to catch it after the EndProcess method is called (A).

QUESTION 43

You are writing an application that reads messages from a message queue. The name of the message queue is stored in a member variable named queueName. When a message is read, the application processes the message. The code for the application is as follows:

```
class MyApp {
MessageQueue queue;
public MyApp()
{
queue = new MessageQueue(queueName, QueueAccessMode.Receive);
queue.ReceiveCompleted += new
ReceiveCompletedEventHandler(this.ReceivedMessage);
queue.BeginReceive();}
bool KeepListening() { }
void ProcessMessage(Message m) { }
}
```

You need to ensure that the application continues to read messages when the KeepListening method returns True, and stops when the KeepListening method returns False.

What should you do?

A. Use the following implementation of the ReceivedMessage method.

```
private void ReceivedMessage(object sender, ReceiveCompletedEventArgs e) { queue =  
new MessageQueue(queueName, QueueAccessMode.Receive);  
ProcessMessage(queue.EndReceive(e.AsyncResult));  
if (KeepListening() == false) {  
queue.ReceiveCompleted -= new ReceiveCompletedEventHandler(this.  
ReceivedMessage);  
}  
}
```

B. Use the following implementation of the ReceivedMessage method.

```
private void ReceivedMessage(object sender, ReceiveCompletedEventArgs e) { queue =  
new MessageQueue(queueName, QueueAccessMode.Receive);  
ProcessMessage(queue.EndReceive(e.AsyncResult));  
if (KeepListening() == true) {  
queue.BeginReceive();  
}  
}
```

C. Use the following implementation of the ReceivedMessage method.

```
private void ReceivedMessage(object sender, ReceiveCompletedEventArgs e) {  
ProcessMessage(queue.EndReceive(e.AsyncResult));  
if (KeepListening() == false) {  
queue.ReceiveCompleted -= new ReceiveCompletedEventHandler(this.  
ReceivedMessage);  
}  
}
```

D. Use the following implementation of the ReceivedMessage method.

```
private void ReceivedMessage(object sender, ReceiveCompletedEventArgs e) {  
ProcessMessage(queue.EndReceive(e.AsyncResult));  
if (KeepListening() == true) {  
queue.BeginReceive();  
}  
}
```

Answer: D

Explanation:

All of the answers correctly show the message signature. Since the constructor of the application creates the messageQueue object to be used for processing messages, neither answer (A), nor answer (B) would be correct, since they recreate it.

Answer (C) correctly reads the message, and then removes the delegate if KeepListening is false. This goes beyond the requirements of the scenario, that states the processing should stop in this case, not disable the ability to receive a message.

Therefore answer (D) makes the most sense, it receives the appropriate message, tests the value for KeepListening, and if it is positive it calls the BeginReceive method to continue reading messages, as required in the scenario.

QUESTION 44

A class library named MathLib contains the following code.

```
public class MathClass : MarshalByRefObject {
public decimal DoHugeCalculation(int iterations)
{
decimal result;
//Some very lengthy calculations
...
return result; }
}
```

The MathLib class is hosted in a .NET Framework remoting server application. A Windows application project running on a client computer contains the following class.

```
public class MathClient { public void ProcessHugeCalculation(int iterations)
{
MathClass cm = new MathClass();
decimal decRes = cm.DoHugeCalculation(iterations);
//process the result
... }
}
```

The MathClient class must call the MathClass class asynchronously by using remoting. A callback must be implemented to meet this requirement. You need to complete the implementation of the MathClient class. What should you do?

A. Apply the Serializable attribute to the MathClient class.

B. Modify the MathClient class as follows:

```
public class MathClient {public delegate void DoHugeCalculationDelegate(decimal
result);
public event DoHugeCalculationDelegate DoHugeCalculationResult;
public void DoHugeCalculationHandler(decimal result)
{DoHugeCalculationResult(result);
}
public void ProcessHugeCalculation(int iterations) {
//Hook up event handler here...
... }
}
```

C. Apply the OneWay attribute to all methods in the MathClass class.

D. Modify the MathClient class as follows:

```
public class MathClient { private delegate decimal DoHugeCalculationDelegate(int
iterations);
private void DoHugeCalculationCallBack(IAsyncResult res) {
AsyncResult aRes = (AsyncResult)res;
decimal decRes = ((DoHugeCalculationDelegate)aRes.
AsyncDelegate).EndInvoke(res);
//process the result
... } public void ProcessHugeCalculation(int iterations) {
```

```
MathClass cm = new MathClass();
DoHugeCalculationDelegate del = new DoHugeCalculationDelegate(
cm.DoHugeCalculation);
del.BeginInvoke(iterations, new AsyncCallback(
DoHugeCalculationCallBack), null); }}
```

Answer: D

Explanation:

The serializable attribute (A) only indicates that a class can be serialized, which does not answer any of the requirements of this scenario.

The OneWay attribute (C) marks methods as one way, without return values or out or ref parameters, none of which would be applicable to this scenario.

The remaining answers attempt to create a delegate, as required in the scenario. In answer (B) the delegate is declared with public scope, something that is not required since it will only be used within the class, while in (D) the delegate is declared properly with private scope.

Answer (B) also declares an event of the delegate type, which is not required, since the scenario requires that a callback be implemented (as in (D)).

Finally, answer (D) also properly specifies the callback, as part of the BeginInvoke method.

QUESTION 45

An application fails when executing a specific operation. You discover that the failure occurs when an exception is raised by a Web service. The application uses the following code to call the Web service.

```
void Process() {
ProcessService serviceProxy = new ProcessService();
serviceProxy.ProcessDataCompleted += new
ProcessDataCompletedEventHandler(ServiceCompleted);
serviceProxy.ProcessDataAsync(data);
}
```

You need to ensure that the application does not fail when the Web service raises the exception. Your solution must maximize the performance of your code.

What should you do?

A. Register the following method with the proxy object to receive the notification of completion.

```
void ServiceCompleted(object sender, ProcessDataCompletedEventArgs e)
{
if (e.Error != null)
LogMessage(e.Error.Message);
else
ProcessResult(e.Result);
}
```

B. Register the following method with the proxy object to receive the notification of

completion.

```
void ServiceCompleted(object sender, ProcessDataCompletedEventArgs e)
{
    if (e.Error is SoapException)
        LogMessage(e.Error.Message);
    else
        ProcessResult(e.Result);
}
```

C. Register the following method with the proxy object to receive the notification of completion.

```
void ServiceCompleted(object sender, ProcessDataCompletedEventArgs e)
{
    if (sender is SoapException)
        LogMessage(e.Error.Message);
    else
        ProcessResult(e.Result);
}
```

D. Register the following method with the proxy object to receive the notification of completion.

```
void ServiceCompleted(object sender, ProcessDataCompletedEventArgs e)
{
    try {
        ProcessResult(e.Result);
    } catch (Exception ex) {
        Console.WriteLine(ex.Message);
    }
}
```

Answer: D

Explanation:

All of the answers create the correctly named method, with the appropriate parameters. The sender in this case would be the Proxy object, which would not be a SoapException so answer (C) is obviously incorrect.

The question then comes down to - when is an exception thrown by the web service received by the client? In this case the Exception will not come to light until ProcessResult(e.Result) is called.

There is no indication in the question that e has an Error property, therefore both (A) and (B) are wrong, and secondly when either subsequently calls the ProcessResult method the exposed Exception will cause the application to fail in contraindication of the requirements of the scenario.

Therefore, the only plausible answer is to sandwich the call to ProcessResult in a try block to catch any Exception received.

Another way to look at it - even if the rational above is not used - would it be faster to check the value a property and then call a method (A), or to cast a property to an object and then call a method (B), or just call the method and see what happens (D)?

QUESTION 46

A Windows service application must host a .NET Framework remoting class named SimpleMathClass. SimpleMathClass must have the following method, which can be called remotely.

```
public: double Add(double x, double y) { return x + y;}
```

You need to create the SimpleMathClass class.

What should you do?

A. Write the following class definition.

```
public ref class SimpleMathClass : public MarshalByRefObject { public: double  
Add(double x, double y) {  
return x + y; }  
};
```

B. Write the following class definition.

```
[Serializable]public ref class SimpleMathClass { public: double Add(double x, double y)  
{  
return x + y; }  
};
```

C. Write the following class definition.

```
public ref class SimpleMathClass : public ServicedComponent { public: double  
Add(double x, double y) {  
return x + y; }  
};
```

D. Write the following class definition.

```
[AutomationProxy(true)]public ref class SimpleMathClass { public: double Add(double  
x, double y) {  
return x + y; } };
```

Answer: A

Explanation:

The MarshalByRefObject class (A) enables access to this class across application domain boundaries, which meets the criterion of the scenario.

The Serializable attribute (B) indicates that a class can be serialized, so that it can be passed to a remote location, but does not indicate whether it can be called remotely.

The ServicedComponent class (C) represents the base class of all classes using COM+ services, but since this application will be hosted in a Windows service this definition would not meet the necessary criteria.

The AutomationProxy attribute (D) specifies whether the class should be marshaled using the Automation marshaler, or a custom proxy and stub, which addresses how the class can be passed, but not necessarily how it can be called.

QUESTION 47

A .NET Framework remoting server hosts a class library that contains the following class.

```
public class SimpleMathClass : MarshalByRefObject { public int LogData(DataRow dr)
```

```
{ //Lengthy database calls ... }  
}
```

Users of a Windows-based client application report that the application often becomes nonresponsive. You discover that the application makes calls to the LogData method that take several seconds to return. The return value is required to generate reports in the client application. You need to ensure that calls to the LogData method can be processed without making the client application nonresponsive.

What should you do?

- A. Call the LogData method by using the ThreadPool.QueueUserWorkItem method.
- B. In the client code, declare a delegate that has the same signature as the LogData method. On the client application's main thread, call the delegate's BeginInvoke method and pass in the necessary data. On a second thread, call the EndInvoke method on the delegate to get the results.
- C. In the client code, declare a delegate that has the same signature as the LogData method. On the client application's main thread, call the delegate's Invoke method and pass in the necessary data. On a second thread, call the GetObjectData method on the delegate to get the results.
- D. Apply the OneWay attribute to the LogData method.

Answer: B

Explanation:

The ThreadPool.QueueUserWorkItem method (A) queues a method for execution, which executes when a thread pool thread becomes available. This scenario doesn't mention thread pooling, so is incorrect.

The OneWay attribute (D) marks a method as one way, without a return value or out or ref parameters, and is used for messaging. This scenario doesn't mention messaging and does have a return value, so is incorrect.

The remaining answers say to declare a delegate that has the same signature as the method. Answer (C) says to call the delegates Invoke method on the main thread, potentially making the client application nonresponsive, which is what we're trying to prevent. Answer (B) is therefore correct - it calls the delegate asynchronously and then requests the results on the secondary thread.

QUESTION 48

A .NET Framework application receives messages from a private message queue named MyQueue. The private queue exists on a computer named APPSERVER1. A user in a domain named CONTOSO has the username dhall. This user reports that when he attempts to run the application, he receives a MessageQueueException exception with the error message text "Access to Message Queuing system is denied." You need to ensure that CONTOSO\dhall can receive messages from MyQueue using the .NET Framework application. Your solution must not give dhall unnecessary privileges. What should you do?

- A. Write the following code segment.


```
MessageQueuePermissionEntry mqpe =  
new MessageQueuePermissionEntry( MessageQueuePermissionAccess.Administer  
, @"APPSERVER1\Private$\MyQueue ", @"contoso\dhall" , null );  
MessageQueuePermission p = new MessageQueuePermission(new  
MessageQueuePermissionEntry[] { mqpe } );  
p.Demand();
```

B. Write the following code segment.

```
MessageQueue q = new MessageQueue(@"APPSERVER1\Private$\MyQueue");  
q.SetPermissions(@"contoso\dhall" , MessageQueueAccessRights.GenericRead);
```

C. Write the following code segment.

```
MessageQueuePermissionEntry mqpe = new MessageQueuePermissionEntry(  
MessageQueuePermissionAccess.Receive , @"APPSERVER1\Private$\MyQueue" ,  
@"contoso\dhall" , null );  
MessageQueuePermission p = new MessageQueuePermission(new  
MessageQueuePermissionEntry[] { mqpe } );  
p.Demand();
```

D. Write the following code segment.

```
MessageQueue q = new MessageQueue(@"APPSERVER1\Private$\MyQueue");  
q.SetPermissions(@"contoso\dhall" , MessageQueueAccessRights.FullControl);
```

Answer: B

Explanation:

The scenario specifically says to not give dhall unnecessary privileges. By looking at the answers, if answers (A) or (D) would work, the access they would provide are contrary to this requirement in that answer (A) provides Administer access to the queue, and (D) provides FullControl to the queue.

What is needed is to grant access to the message queue for this user - not to get or set its permissions. Using the MessageQueuePermission object allows control of code access permissions for messaging; not access to a given queue, so neither (A), nor (C) provide solutions for the requirements.

This leaves answer (B), which creates a MessageQueue class that provides access to a queue on a Message Queuing server, and sets the Access Rights to GenericRead, all in keeping with the requirements of this scenario.

QUESTION 49

A Console Application calls a Web service named SessionStateService five times, sequentially. The IncrementSessionCounter Web service method increments and returns an integer value that is held in a cookie. (Line numbers are included for reference only.)

```
01  
void IncrementSession5Times() {  
02  
SessionStateService service = null;  
03  
04  
for (int i = 0; i < 5; i++) {
```

```
05
06
Console.WriteLine("Loop Pass {0} - result = {1}",
i,
08
service.IncrementSessionCounter());
09
}
10
}
```

You need to ensure that when the IncrementSession5Times method is run, the following output is displayed.

Loop Pass 0 - result = 1Loop Pass 1 - result = 2Loop Pass 2 - result = 3Loop Pass 3 -
result = 4Loop Pass 4 - result = 5

What should you do?

A. Replace line 03 with the following code.

```
CookieContainer cookies = new CookieContainer();
```

Replace line 05 with the following code.

```
SessionStateService service = new SessionStateService();
service.CookieContainer = cookies;
```

B. Replace line 03 with the following code.

```
SessionStateService service = new SessionStateService();
```

C. Replace line 05 with the following code.

```
SessionStateService service = new SessionStateService();
CookieContainer cookies = new CookieContainer();
service.CookieContainer = cookies;
```

D. Replace line 03 with the following code.

```
CookieContainer cookies = new CookieContainer();
Replace line 05 with the following code.SessionStateService service = new
SessionStateService();cookies.GetCookies(new Uri(service.Url));
```

Answer: D

Explanation:

In order for the SessionsStateService to be called five times it has to be called from within the loop. This eliminates answer (B) as a possibility.

There only needs to be one CookieContainer to provide storage for cookies, therefore answer (C) cannot be the answer.

There is not indication in the question that the service has a CookieContainer property to which cookies can be assigned, so answer (A) is questionable.

The CookieContainer class does, however, have a GetCookies method that takes a URI value that gets a CookieCollection containing the Cookie instances associated with the URI, which is what we want in this scenario, so answer (D) is correct.

QUESTION 50

You are writing an application that provides a graphical administrative interface to a message queue that displays information about the messages. The GUI is updated in the AdminFunction function. You need to ensure that the GUI is updated without unnecessary delay, and that processed messages contain the most recent message contents. What should you do?

A. Read the messages from the message queue and update the GUI by using the following code segment.

```
MessageQueue queue = new MessageQueue(queueName, QueueAccessMode.Peek);
Cursor queueCursor = queue.CreateCursor();
Message m = queue.Peek(timeout, queueCursor, PeekAction.Current);
while (m != null) { AdminFunction(m);
m = queue.Peek(timeout, queueCursor, PeekAction.Next);
}
```

B. Read the messages from the message queue and update the GUI by using the following code segment.

```
MessageQueue queue = new MessageQueue(queueName,
QueueAccessMode.Peek);
Cursor cursor = queue.CreateCursor();
Message m = queue.Peek();
while (m != null) { AdminFunction(m);
m = queue.Peek();
}
```

C. Read the messages from the message queue and update the GUI by using the following code segment.

```
MessageQueue queue = new MessageQueue(queueName,
QueueAccessMode.Peek);
Message[] messages = queue.GetAllMessages();
for (int i = 0; i < messages.Length; i++)
{
AdminFunction(messages[i]);
}
```

D. Read the messages from the message queue and update the GUI by using the following code segment.

```
MessageQueue queue = new MessageQueue(queueName,
QueueAccessMode.Peek);
MessageEnumerator messages = queue.GetMessageEnumerator();
while (messages.MoveNext())
{
AdminFunction(messages.current);
}
```

Answer: D

Explanation:

Answer (A) is invalid because the queue.Peek method does not accept parameters, and only removes the object at the beginning of the Queue without removing it from the Queue. In addition, there is no CreateCursor method on the queue object.

The declaration of a Cursor in answer (B) makes this answer wrong, simply because there is no Cursor class that works for iteration, and secondly because it is not used in the remaining code.

There is no queue.GetAllMessages method, so answer (C) is invalid.

Therefore, answer (D) has to be correct. The GetEnumerator method of the queue object returns an enumerator that iterates through the Queue, but does not have exclusive access to the collection.

QUESTION 51

You write a .NET Framework remoting application that broadcasts messages to client computers through a central server by raising events on the client computers. Message details are contained in an argument in the event delegate, as shown in the following code segment.

```
public delegate void MessageArrivedHandler(BroadcastEventArgs args);  
public class BroadcastEventArgs {  
    public string Message;  
    public string Sender;  
    public DateTime TimeSent;  
}
```

You need to ensure that the client computer can access the message details contained in the event argument. Your solution cannot change the common language runtime (CLR) security restrictions. What should you do?

- A. Apply the Serializable attribute to the BroadcastEventArgs object.
- B. Change the BroadcastEventArgs class so that it extends the MarshalByRefObject object.
- C. Add the well-known type named BroadcastEventArgs in the server and configure the WellKnownObjectMode mode as SingleCall.
- D. In the client application's remoting configuration file, make the BroadcastEventArgs class a client-activated object (CAO) that points to the server.

Answer: A

Explanation:

In this scenario, the BroadcastEventArgs is passed as part of a remote method call. By applying the Serializable attribute (A) to this class it makes it possible to serialize this class, making it possible to pass the object as part of a remote method call - thus, this is the correct answer.

The MarshalByRefObject (B) enables access to objects across application domain boundaries for applications that support remoting. It is designed for objects that communicate across application domain boundaries by exchanging messages using a proxy. This scenario does not imply a proxy.

Since the scenario has the server raising events on the client, neither a Server activated object (C), nor a client activated object (D), both of which would be instantiated on the Server make sense.

QUESTION 52

You are writing a .NET Framework remoting client application that must call two remoting servers. The first server hosts an assembly that contains the following delegate and class definition.

```
public delegate bool IsValidDelegate(string number, Int16 code);
public class CreditCardValidator : MarshalByRefObject
{ public bool IsValid (string number, Int16 code) {
//some data access calls that are slow under heavy load
... }
}
```

The second server hosts an assembly that contains the following delegate and class definition.

```
public delegate float GetCustomerDiscountDelegate( int customerId);
public class PreferredCustomer { public float GetCustomerDiscount(int customerId) {
//some data access calls that are slow under heavy load
... }
}
```

You configure the remoting client application to call both server classes remotely. The amount of time it takes to return these calls varies, and long response times occur during heavy load times. The processing requires the result from both calls to be returned. You need to ensure that calls to both remoting servers can run at the same time. What should you do?

A. Write the following code segment in the client application.

```
PreferredCustomer pc = new PreferredCustomer();
GetCustomerDiscountDelegate del1 = new
GetCustomerDiscountDelegate(pc.GetCustomerDiscount);
IAsyncResult res1 = del1.BeginInvoke(1001, null, null);
CreditCardValidator val = new CreditCardValidator();
IsValidDelegate del2 = new IsValidDelegate( val.IsValid);
IAsyncResult res2 = del2.BeginInvoke("4111-2222-3333-4444" , 123, null, null);
WaitHandle[] waitHandles = new WaitHandle[]
{res1.AsyncWaitHandle, res2.AsyncWaitHandle};
ManualResetEvent.WaitAll(waitHandles);
double discount = del1.EndInvoke(res1);
bool isValid = del2.EndInvoke(res2);
```

B. Write the following code segment in the client application.

```
PreferredCustomer pc = new PreferredCustomer();
GetCustomerDiscountDelegate del1 = new
GetCustomerDiscountDelegate(pc.GetCustomerDiscount);
CreditCardValidator val = new CreditCardValidator();
IsValidDelegate del2 = new IsValidDelegate(val.IsValid);
double discount = del1.Invoke(1001);
bool isValid = del2.Invoke("4111-2222-3333-4444", 123);
```

C. Write the following code segment in the client application.

```
PreferredCustomer pc = new PreferredCustomer();
```

```
GetCustomerDiscountDelegate del1 = new
GetCustomerDiscountDelegate(pc.GetCustomerDiscount);
IAsyncResult res1 = del1.BeginInvoke(1001, null, null);
double discount = del1.EndInvoke(res1);
CreditCardValidator val = new CreditCardValidator();
IsValidDelegate del2 = new IsValidDelegate( val.IsValid);
IAsyncResult res2 = del2.BeginInvoke("4111-2222-3333-4444", 123, null, null);
bool isValid = del2.EndInvoke(res1);
D. Write the following code segment in the client application.PreferredCustomer pc =
new PreferredCustomer();
CreditCardValidator val = new CreditCardValidator();
double discount = pc.GetCustomerDiscount(1001);
bool isValid = val.IsValid("4111-2222-3333-4444", 123);
```

Answer: A

Explanation:

Answer (D) since it makes synchronous calls to both methods that can have varied response times is obviously false.

Despite the instantiation of the two delegates in answer (B) they are both synchronously invoked, so this is not a correct answer.

In answer (C) the first delegate is invoked asynchronously, but then immediately the EndInvoke method is called, which will block the application processing until the method returns, which makes this an inefficient use of threading, and incorrect.

In answer (A) the two delegates are instantiated and invoked asynchronously, and then a WaitHandle is created to ensure that both methods have been returned before the EndInvoke methods are called which meets the requirements of this scenario.

QUESTION 53

You are creating a Web service to expose the public methods on a class. Two overloaded methods are defined in the class as follows:

```
public Customer GetCustomer(string custId)
public Customer GetCustomer(string name, string postalCode)
```

You need to expose both methods on the Web service. Which code segment should you use?

A. [WebMethod][SoapDocumentMethod(Action="GetCustomerById")]public Customer GetCustomer(string custId){ ... }

[WebMethod][SoapDocumentMethod(Action="GetCustomerByName")]public Customer GetCustomer(string name, string postalCode){ ... }

B. [WebMethod][SoapDocumentMethod(RequestElementName="GetCustomerById")]public Customer GetCustomer(string custId){ ... }

[WebMethod][SoapDocumentMethod(RequestElementName="GetCustomerByName")]public Customer GetCustomer(string name, string postalCode){ ... }

C. [WebMethod(Description="GetCustomerById")]public Customer GetCustomer(string

```
custId){ ... } [WebMethod(Description="GetCustomerByName")]public Customer  
GetCustomer(string name, string postalCode){ ... }  
D. [WebMethod(MessageName="GetCustomerById")]public Customer  
GetCustomer(string custId){ ... }  
[WebMethod(MessageName="GetCustomerByName")]public Customer  
GetCustomer(string name, string postalCode){ ... }
```

Answer: D

Explanation:

The only way to expose overloaded methods on a Web service is by using the MessageName property of the WebMethod attribute, thus answer (D) is correct.

The SoapDocumentMethod attribute specifies that SOAP messages to and from the method use Document formatting, or according to an XSD schema, which has nothing to do with allowing overloading, so (A) and (B) are incorrect.

The Description property of the WebMethod attribute just provides a descriptive message describing the method, so does not do anything about overloading, and so is incorrect.

QUESTION 54

A .NET Framework remoting server hosts a class library that contains the following class.

```
public class SimpleMathClass : MarshalByRefObject  
{  
    public void LogData(DataRow dr)  
    { //Lengthy database calls ... }  
}
```

Users of a Windows-based client application report that the application often becomes nonresponsive. You discover that the application makes calls to the LogData method that take several seconds to return. You need to ensure that calls to the LogData method can be processed without causing the client application to become nonresponsive.

What should you do?

- A. Apply the SoapRpcMethodAttribute attribute to the LogData method and initialize its OneWay member to True.
- B. Modify the method so that it is an anonymous method.
- C. Apply the OneWayAttribute attribute to the LogData method.
- D. Modify the class to implement the IAsyncResult interface.

Answer: C

Explanation:

Using a OneWay attribute (C) allows a method to be invoked, with no reply message or status or information expected. Since LogData returns void and has only an in parameter it qualifies for this attribute, and the application will immediately continue after calling the method as is required in this scenario.

Applying the SoapRpcMethod attribute (A) specifies that SOAP messages sent to and

from the method use RPC formatting, but the scenario does not imply the use of SOAP, so this answer is incorrect.

An anonymous method (B) is a way to pass a code block as a delegate parameter when having to create a method might seem unnecessary overhead. This scenario would not be helped by implementing a delegate, either regular or anonymous.

The IAsyncResult interface (D) represents the status of an asynchronous operation, and is the return type of methods that initiate an asynchronous operation, which does not fit with this scenario, and is incorrect.

QUESTION 55

A Windows service application must host a .NET Framework remoting class named SimpleMathClass. SimpleMathClass must have the following method, which can be called remotely.

```
public double Add(double x, double y) { return x + y;}
```

You need to create the SimpleMathClass class.

What should you do?

A. Write the following class definition.

```
[AutomationProxy(true)]public class SimpleMathClass
{ public double Add(double x, double y)
{
return x + y; }
}
```

B. Write the following class definition.

```
[Serializable]public class SimpleMathClass
{ public double Add(double x, double y)
{
return x + y; }
}
```

C. Write the following class definition.

```
public class SimpleMathClass : MarshalByRefObject
{ public double Add(double x, double y)
{
return x + y; }
}
```

D. Write the following class definition.

```
public class SimpleMathClass : ServiceComponent
{ public double Add(double x, double y)
{
return x + y; }
}
```

Answer: C

Explanation:

The AutomationProxy attribute (A) specifies that this class should be marshaled using the Automation marshaler (also known as the type library marshaler or universal marshaler),

which means that the interface does not need to have a proxy and a stub. Since the scenario calls for a remoting system, this does not apply.

Adding the Serializable attribute (B) indicates that the class could be serialized. Since only public and private fields marked this way are serialized, so adding it to the class in order to call the Add method does not meet the requirements of the scenario.

Inheriting from the MarshalByRefObject class enables access to objects across application domain boundaries, thus calls to methods remotely are available, and what is sought in this scenario.

Inheriting from the ServicedComponent class enables a class to be used with COM+ services, and does not meet the requirements of this scenario.

QUESTION 56

A message queue named SecureQueue requires incoming messages to be authenticated. When an application attempts to send a message to SecureQueue, the following exception is thrown. User's internal Message Queuing certificate does not exist. The following code is used to send the message. (Line numbers are included for reference only.)

```
01 MessageQueue mq = new MessageQueue(".\SecureQueue");
```

```
02 Message m = new Message("Test Message");
```

```
03 m.UseAuthentication = true; 04 mq.Send(m);
```

You need to ensure that a message can be sent to SecureQueue without the exception being thrown. What should you do?

A. Insert the following line of code between lines 03 and

```
04.m.AuthenticationProviderType = CryptographicProviderType.Rsqsig;
```

B. Replace line 03 with the following line of code. m.EncryptionAlgorithm = EncryptionAlgorithm.Rc4;

C. Replace line 03 with the following line of code. m.AttachSenderId = true;

D. Insert the following line of code between lines 03 and

```
04.m.AuthenticationProviderName = "Rsqsig";
```

Answer: C

Explanation:

The idea of this question is to make sure that an exception is not thrown.

The Message.AuthenticationProviderType property (A) gets or sets the type of cryptographic provider used to generate the digital signature of the message, so has nothing to do with this problem.

The EncryptionAlgorithm property (B) gets or sets the encryption algorithm used to encrypt the body of a private message, but does not affect whether an exception is thrown.

The AttachSenderId property (C) gets or sets whether the sender identifier should be attached to the message. In this case it is attached, and is used by the receiving Queue Manager to verify whether the sender has access rights to SecureQueue, which is what we're looking for.

The AuthenticationProviderName property (D) gets or sets the name of the cryptographic provider used to generate the digital signature of the message.

QUESTION 57

When you execute a client application, the following exception is thrown. `EncryptedKeyToken` is expected but not present in the security header of the incoming message. You discover that the exception is thrown when the client application invokes a Web service named `Math` with the following code. (Line numbers are included for reference only.)

```
01 try
02 {
03     MathWse^ ws = gcnew MathWse ();
04
05     int result = ws->Add(1, 2);
06 }
07 catch (Exception^ ex)
08 {
09     MessageBox::Show(ex->Message);
10 }
```

The client application and Web service have the same Web Services Enhancements (WSE) 3.0 policy. The policy configuration file contains the following policy section.

```
<policy name="Secure">
<anonymousForCertificateSecurity
establishSecurityContext="false"
renewExpiredSecurityContext="true"
requireSignatureConfirmation="false"
messageProtectionOrder="SignBeforeEncrypt"
requireDerivedKeys="true"
ttlInSeconds="300">
<!-- XML defining the serviceToken and protection -->
</anonymousForCertificateSecurity>
</policy>
```

You need to ensure that the client application can communicate with the Web service. What should you do?

- A. Add the following code between lines 03 and 04. `ws->SetPolicy("Secure");`
- B. Replace line 03 with the following code. `Math^ ws = gcnew Math();`
- C. Add the following code between lines 03 and 04. `UsernameToken^ u = gcnew UsernameToken("userid", "password"); EncryptedKeyToken^ et = gcnew EncryptedKeyToken(u); ws->SetClientCredential(et);`
- D. Add the following code between lines 03 and 04. `ws->UseDefaultCredentials = true;`
- E. Add the following code between lines 03 and 04. `ws->SetPolicy("anonymousForCertificateSecurity");`

Answer: A

Explanation:

There are two techniques for applying a policy with a client; the first is to call the SetPolicy method of a WSE 3.0-generated proxy class, and the other is to annotate the proxy class with a Policy attribute. So the answer would look like either ws.SetPolicy("Secure"), or a class definition that looks like [Policy(typeof(Secure))] ... public partial class MathServiceWse : WebServicesClientProtocol . . .

Obviously, answer is (A) calls the SetPolicy method on the proxy and passes in the appropriate security, and none of the other answers are close.

NOTE: This question is supposed to be in C#, not in C++.

QUESTION 58

You are creating a Web service. You need to add an operation that can be called without returning a message to the caller. Which code should you use?

- A. [WebMethod][SoapDocumentMethod(OneWay = true)]public void ProcessName(string name) { ... }
- B. [WebMethod][OneWay()]public void ProcessName(string name) { ... }
- C. [WebMethod][SoapDocumentMethod(Action = "OneWay")]public void ProcessName(string name) { ... }
- D. [WebMethod][SoapDocumentMethod(OneWay = true)]public bool ProcessName(string name) { ... return false; }

Answer: A

Explanation:

The SoapRpcMethod and SoapDocumentMethod attributes both have a OneWay property. In addition, there is a separate OneWay attribute, but OneWay is not a possible value for the Action property of the SoapDocumentMethod attribute (C), so this answer is incorrect.

In order to create one-way method it has to return void, and only have in parameters, it cannot throw any exceptions, and ref and out parameters are not supported. These requirements eliminate (D) since it has a bool return type.

The syntax for both (A) and (B) is correct, the difference is that the SoapDocumentMethod attribute is used with Web Services, while the OneWay attribute is used in remoting, So (A) is the most correct answer for this scenario.

QUESTION 59

A client application calls a Web service named Math. Both the client application and Math are configured with a Web Services Enhancements (WSE) 3.0 policy named Secure to validate anonymous access for certificate security. A Web reference to the Math Web service is added to the client application's project using Microsoft Visual Studio 2005. When the client application is built and executed, a SoapException exception is thrown with the following message. The security requirements are not met because the security header is not included in the incoming message. You need to ensure that the application runs without throwing the SoapException exception. What should you do?

- A. Modify the Math proxy class so that it derives from the following protocol.`System::Web::Services::Protocols::SoapHttpClientProtocol`
- B. Add the following attribute to the Math proxy class definition.
`[Microsoft::Web::Services3::Policy("Secure")]`
- C. Add the following attribute to the Math proxy class definition.
`[Microsoft::Web::Services3::Policy("anonymousForCertificateSecurity")]`
- D. Set the Name property for the `WebServiceBindingAttribute` attribute on the Math proxy class definition to `MathWseSoap`.

Answer: B

Explanation:

A WSE 3.0 class inherits from the `WebServicesClientProtocol` class not the `SoapHttpClientProtocol` class (A), so this answer is incorrect.

Since the idea is to require the policy to applied to the proxy class, adding the `Policy` attribute, and specifying the correct policy name will meet the requirements of the question. In this case (B) identifies the correct policy and is correct, while (C) does not, and is not.

The `WebServiceBinding` attribute (D) declares a binding (similar to an interface) that defines one or more XML Web Service methods. It has nothing to do with security, and is therefore not the answer.

NOTE: The code for this question is supposed to be in C#.

QUESTION 60

An assembly named `SimpleMathLib` is deployed to the `Bin` folder that is under a virtual directory named `SimpleMathHost`, which is on an IIS server named `SERVER1`. The `SimpleMathLib` assembly contains the following code.

```
Public Class SimpleMathClass Inherits MarshalByRefObject
Public Function Add(ByVal x As Double, ByVal y As Double) _ As Double
Return x + y
End Function
End Class
```

The `Web.config` file under the `SimpleMathHost` virtual directory contains the proper configuration to host `SimpleMath` as a remoting object. You write a client Console Application and add a reference to the `SimpleMathLib` assembly. You need to ensure that the client Console Application calls the `Add` method on `SERVER1` and returns the correct sum of the parameters to the Console Application. What should you do?

- A. Write the following code in the client application.

```
Dim sm As SimpleMathClass =
Type(Activator.CreateInstance(GetType(SimpleMathClass),
_ New String() {"http: //server1:80/SimpleMathHost/SimpleMath.rem"}, _
"SimpleMath.SimpleMathClass, SimpleMathLib"
}), _ SimpleMathClass)
Console.WriteLine(sm.Add(2, 2).ToString())
```

A. Write the following code in the client application.

```
Dim sm As SimpleMathClass = _  
CType(Activator.GetObject(GetType(SimpleMathClass),  
_ "http://server1:80/SimpleMathHost/SimpleMath.rem"), SimpleMathClass)  
Console.WriteLine(sm.Add(2, 2).ToString())
```

B. Write the following code in the client application.

```
Dim chan As New TcpChannel()  
ChannelServices.RegisterChannel(chan, False)  
Dim sm As New SimpleMathClass()  
Console.WriteLine(sm.Add(2, 2).ToString())
```

C. Write the following code in the client application.

```
Dim sm As SimpleMathClass = _  
CType(Activator.GetObject(GetType(SimpleMathClass),  
_ "tcp://server1:80/SimpleMathHost/SimpleMath.rem"), SimpleMathClass)  
Console.WriteLine(sm.Add(2, 2).ToString())
```

Answer: B

QUESTION 61

You are writing an installation application for a Windows Forms application. The Windows Forms application requires a private message queue named MyQueue. You need to ensure that the message queue exists after installation. What should you do?

A. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer class.

```
Dim q As MessageQueue =  
MessageQueue.Create(".\Private$\MyQueue")
```

B. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer class.

```
Dim q As New Queue(Of  
Message)q.Enqueue(New Message(".\Private$\MyQueue"))
```

C. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer class.

```
Dim q As New  
MessageQueue(".\Private$\MyQueue")
```

D. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer class.

```
Dim q As New  
Queue()q.Enqueue(".\Private$\MyQueue")
```

Answer: A

QUESTION 62

A file named Util.asmx contains the following code segment. (Line numbers are included for reference only.)

```
01  
<%@ WebService Language="VB" Class="Util" %>  
02  
Public Class Util  
03
```

```
Public Function GetData() As String
```

```
04
```

```
Return "data"
```

```
05
```

```
End Function
```

```
06 End Class
```

You need to expose the GetData method through a Web service. What should you do?

A. Insert the following line of code between lines 02 and

03.<System.Web.Services.WebMethod(>_

B. Insert the following line of code between lines 02 and 03.Inherits

System.Web.Services.WebService

C. Replace line 01 with the following line of code.<%@ WebService Language="VB"

Class="System.Web.Services.WebService" %>

D. Insert the following line of code between lines 01 and

02.<System.Web.Services.WebService(>_

Answer: A

QUESTION 63

A Web service application uses Web Services Enhancements (WSE) 3.0. A class named RejectUnknownActorFilter is derived from the SoapFilter class. The RejectUnknownActorFilter class throws a SoapException exception if the request contains an unexpected actor. A class defines a policy assertion as follows. (Line numbers are included for reference only.)

```
01
```

```
Public Class RequireActorAssertion
```

```
02
```

```
Inherits PolicyAssertion
```

```
03
```

```
Public Overrides Function CreateClientInputFilter(ByVal _  
context As FilterCreationContext) As SoapFilter
```

```
04
```

```
Return Nothing
```

```
05
```

```
End Function
```

```
06
```

```
Public Overrides Function CreateClientOutputFilter(ByVal _  
context As FilterCreationContext) As SoapFilter
```

```
08  
Return Nothing
```

```
09
```

```
End Function
```

```
10
```

```
11
```

```
Public Overrides Function CreateServiceInputFilter(ByVal _  
context As FilterCreationContext) As SoapFilter
```


Return Nothing

13

End Function

14

15

Public Overrides Function CreateServiceOutputFilter(ByVal _
context As FilterCreationContext) As SoapFilter16

Return Nothing

17

End Function

18

End Class

You need to ensure that the Web service rejects any SOAP request that contains an unexpected actor. Your code must minimize the server resources used to process the request. What should you do?

- A. Replace line 08 with the following line of code.
Return New RejectUnknownActorFilter()
- B. Replace line 12 with the following line of code.
Return New RejectUnknownActorFilter()
- C. Replace line 16 with the following line of code.
Return New RejectUnknownActorFilter()
- D. Replace line 04 with the following line of code.
Return New RejectUnknownActorFilter()

Answer: B

QUESTION 64

A Windows service application must host a .NET Framework remoting class named SimpleMathClass. SimpleMathClass must have the following method, which can be called remotely.

Public Function Add(ByVal x As Double, ByVal y As Double) _ As Double

Return x + y

End Function

You need to create the SimpleMathClass class.

What should you do?

- A. Write the following class definition.

```
<AutomationProxy(True)> _Public Class SimpleMathClass
Public Function Add(ByVal x As Double, ByVal y As Double) _
As Double
Return x + y
End Function
End Class
```

- B. Write the following class definition.

```
Public Class SimpleMathClass Inherits MarshalByRefObject
```

```
Public Function Add(ByVal x As Double, ByVal y As Double) _  
As Double  
Return x + y  
End Function  
End Class
```

C. Write the following class definition.

```
Public Class SimpleMathClass Inherits ServicedComponent  
Public Function Add(ByVal x As Double, ByVal y As Double) _  
As Double  
Return x + y  
End Function  
End Class
```

D. Write the following class definition.

```
<Serializable(> _Public Class SimpleMathClass  
Public Function Add(ByVal x As Double, ByVal y As Double) _  
As Double  
Return x + y  
End Function  
End Class
```

Answer: B

Explanation:

The MarshalByRefObject class (B) enables access to this class across application domain boundaries, which meets the criterion of the scenario.

The Serializable attribute (D) indicates that a class can be serialized, so that it can be passed to a remote location, but does not indicate whether it can be called remotely.

The ServicedComponent class (C) represents the base class of all classes using COM+ services, but since this application will be hosted in a Windows service this definition would not meet the necessary criteria.

The AutomationProxy attribute (A) specifies whether the class should be marshaled using the Automation marshaler, or a custom proxy and stub, which addresses how the class can be passed, but not necessarily how it can be called.

QUESTION 65

You are creating a Web service. The Web service must be configured to receive the following message.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
<soap:Body>  
<givenName xmlns="urn:SampleNS">given name</givenName>  
<surname xmlns="urn:SampleNS">surname</surname>  
</soap:Body>  
</soap:Envelope>
```

You need to ensure that the Web Services Description Language (WSDL) for the Web service describes the message. What should you do?

- A. Write the following code for the Web method.<WebMethod(>
_<SoapDocumentMethod(ParameterStyle:=SoapParameterStyle.Wrapped,
Use:=SoapBindingUse.Encoded)> _Public Function HelloWorld(ByVal firstName As
String, ByVal lastName As String) ...End Function
- B. Write the following code for the Web method.<WebMethod(>
_<SoapDocumentMethod(ParameterStyle:=SoapParameterStyle.Bare,
Use:=SoapBindingUse.Encoded)> _Public Function HelloWorld(ByVal firstName As
String, ByVal lastName As String) ...End Function
- C. Write the following code for the Web method.<WebMethod(>
_<SoapDocumentMethod(ParameterStyle:=SoapParameterStyle.Bare,
Use:=SoapBindingUse.Literal)> _Public Function HelloWorld(ByVal firstName As
String, ByVal lastName As String) ...End Function
- D. Write the following code for the Web method.<WebMethod(>
_<SoapDocumentMethod(ParameterStyle:=SoapParameterStyle.Wrapped,
Use:=SoapBindingUse.Literal)> _Public Function HelloWorld(ByVal firstName As
String, ByVal lastName As String) ...End Function

Answer: C

QUESTION 66

An application has components named ComponentA, ComponentB, and ComponentC. ComponentA and ComponentB update tables in a database named DB1. ComponentC updates tables in a database named DB2. At run time, ComponentA is executed with either ComponentB or ComponentC in a single transaction. You need to compose the transaction to minimize the use of the system resources. What should you do?

- A. Write the following code for the transaction.
<AutoComplete> _Sub ExecuteTransactions() ...
End Sub
- B. Write the following code for the transaction.
Sub ExecuteTransactions()
Try
ContextUtil.SetComplete()
Catch
ContextUtil.SetAbort()
End TryEnd Sub
- C. Write the following code for the transaction.
Sub ExecuteTransactions() Dim connection As SqlConnection = New SqlConnection(...)
Dim trans as SqlTransaction = connection.BeginTransaction();
Try
trans.Commit()
Catch
trans.Rollback()
End TryEnd Sub
- D. Write the following code for the transaction.
Sub ExecuteTransactions() Using txscope As TransactionScope = new

```
TransactionScope()  
txscope.Complete() End Using}
```

Answer: D

QUESTION 67

A Windows-based application sends messages to the TestMessageQueue queue. An acknowledgement message must be sent to the TestMessageQueueAdmin queue when: The original message is retrieved from TestMessageQueue. The message has not been retrieved from TestMessageQueue after a specified time period. The following code is used to send a message to the TestMessageQueue queue. (Line numbers are included for reference only.)

```
01 Dim mq As New MessageQueue(".\TestMessageQueue")  
02 Dim mqAdmin As New MessageQueue(".\TestMessageQueueAdmin")  
03 Dim m As New Message(messageBody)  
04 m.AcknowledgeType = AcknowledgeTypes.PositiveReceive  
05 mq.Send(m)
```

You need to ensure that an acknowledgement message is sent to the TestMessageQueueAdmin queue as required.

Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Replace line 04 with the following code.
m.AcknowledgeType = AcknowledgeTypes.PositiveReceive Or _ AcknowledgeTypes.NegativeReceive
- B. Add the following line of code between lines 04 and 05.
m.ResponseQueue = mqAdmin
- C. Add the following line of code between lines 04 and 05.
m.AdministrationQueue = mqAdmin
- D. Replace line 04 with the following code.
m.AcknowledgeType = AcknowledgeTypes.PositiveReceive And _ AcknowledgeTypes.NegativeReceive

Answer: A,C

QUESTION 68

You are writing an application that handles the batch processing of user accounts. The application assigns network identities for users by calling the following Web service method.

```
<WebMethod(> _Public Function GetNetworkID(ByVal name As String) As String  
...End Function
```

The application calls the Web service using the following code. (Line numbers are included for reference only.)

```
01  
Private Sub ProcessPeople(ByVal people As List(Of Person))  
02  
Dim serviceProxy As PersonService = New PersonService()  
03
```

```
AddHandler serviceProxy.GetNetworkIDCompleted, _
04
AddressOf GetNeworkIDCompleted
05
Dim i As Integer
06
For i = 0 To people.Count - 1
08
Next
09
End Sub
10
11
Private Sub GetNeworkIDCompleted(ByVal sender As Object, _
12
ByVal e As GetNetworkIDCompletedEventArgs)
13
Dim p As Person = Nothing
14
15
p.NetworkID = e.Result
16
ProcessPerson(p)
17
End Sub
```

You need to ensure that the application can use the data supplied by the Web service to update each Person instance. Which two actions should you perform? (Each correct answer presents part of the solution. Choose two.)

- A. Replace line 14 with the following code segment.p = e.UserState
- B. Replace line 14 with the following code segment.p = sender
- C. Replace line 07 with the following code
segment.serviceProxy.GetNetworkIDAsync(people(i).FirstName, people(i))
- D. Replace line 07 with the following code
segment.serviceProxy.GetNetworkIDAsync(people(i).FirstName, Nothing)

Answer: A,C

QUESTION 69

A Web service named Math uses Web Service Enhancements (WSE) 3.0 policy assertions that are defined in the policy configuration file to implement security. The Web service authenticates access to the Web methods using the UsernameOverTransportAssertion policy assertion. A client application must call a Web method on the Math Web service. When the client application is executed, it throws an InvalidOperationException exception that displays the following text.Unable to determine client token to use. Client token type requested was

'Microsoft.Web.Services3.Security.Tokens.UsernameToken'. The following code is used to invoke the Web method. (Line numbers are included for reference only.)

```
01 Math^ ws = gnew Math();
```

```
02 ws->SetPolicy("Secure");
```

```
03 int result = ws->Add(3, 4);
```

You need to ensure that the Web methods can be invoked without causing the client application to throw the InvalidOperationException exception. What should you do?

A. Add the following code between lines 02 and

```
03.ws->SetServiceCredential<UsernameToken^>(gnew UsernameToken(userId, password));
```

B. Add the following code between lines 02 and

```
03.ws->SetClientCredential<UsernameToken^>(gnew UsernameToken(userId, password));
```

C. Add the following code between lines 02 and 03.ws->Credentials =

```
System::Net::CredentialCache::DefaultCredentials;
```

D. Add the following code between lines 02 and 03.ws->Credentials = gnew

```
System::Net::NetworkCredential(userId, password);
```

Answer: B

QUESTION 70

You are writing an installation application for a Windows Forms application. The Windows Forms application requires a private message queue named MyQueue. You need to ensure that the message queue exists after installation. What should you do?

A. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer

```
class.System::Collections::Generic::Queue<Message^> q = gnew
```

```
System::Collections::Generic::Queue<Message^>();
```

```
q.Enqueue(gnew Message(path));
```

B. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer class.MessageQueue^ q = gnew

```
MessageQueue(".\\Private$\\MyQueue");
```

C. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer class.MessageQueue^ q =

```
MessageQueue::Create(".\\Private$\\MyQueue");
```

D. Add an Installer class to the Windows Forms application project. Write the following code in the AfterInstall event for the Installer class.System::Collections::Queue^ q =

```
gnew System::Collections::Queue();
```

```
q->Enqueue(".\\Private$\\MyQueue");
```

Answer: C

QUESTION 71

You create a Web service that exposes a Web method named CalculateStatistics. The response returned by the CalculateStatistics method for each set of input parameters changes every 60 seconds. You need to ensure that all requests to the CalculateStatistics method that have the same set of input parameters, and that occur within a 60-second time period, calculate the statistics only once. Which code segment should you use?

- A. `public:[WebMethod()]String ^CalculateStatistics (array<int> ^values) {
HttpContext::Current->Response->Cache->SetExpires(
DateTime::Now.AddSeconds(60));
...}`
- B. `public:[WebMethod()]String ^CalculateStatistics (array<int> ^values) {
HttpContext::Current->Response->Cache->SetCacheability(
HttpCacheability::Public, "max-age=60");
...}`
- C. `public:[WebMethod(CacheDuration=60)]String ^CalculateStatistics(array<int>
^values) { ... }`
- D. `public:[WebMethod(BufferResponse=60)]String ^CalculateStatistics(array<int>
^values) { ... }`

Answer: C

QUESTION 72

You are writing an application that creates network accounts for users. The application monitors a private queue and it creates network accounts when messages arrive in that queue. A message contains a User object. The message is sent to the queue using the following code.

```
MessageQueue^ q = gcnew MessageQueue(queueName);
Message^ m = gcnew Message(gcnew User("anna", "bedecs"));
q->Send(m, "create account");
```

You need to ensure that the application can process the User object. What should you do?

- A. Write the following code segment to get the User object.
`q->ReceiveById("User");
Message^ m = q->Receive();
User^ u = dynamic_cast<User^>(m->Body);`
- B. Write the following code segment to get the User object.
`q->ReceiveById("User");
m = q->Receive();
m->BodyType = 5;
User^ u = dynamic_cast<User^>(m->Body);`
- C. Write the following code segment to get the User object.
`q->Formatter = gcnew XmlMessageFormatter();
m = q->Receive();
User^ u = dynamic_cast<User^>(m->Body);`

D. Write the following code segment to get the User object.

```
array<Type^>^ types = {User::typeid};
q->Formatter = gcnew XmlMessageFormatter(types);
Message^ m = q->Receive();
User^ u = dynamic_cast<User^>(m->Body);
```

Answer: D

QUESTION 73

You are writing an application that calls a Web service. The application must call the Web service asynchronously and also perform a small amount of processing while the Web service is running. The return value from the Web service is required for additional processing. You need to ensure that the application can call the Web service asynchronously and also process the return value. Your solution must keep processor cycles to a minimum. What should you do?

A. Implement the Web service call as follows:

```
ProcessService^ serviceProxy = gcnew ProcessService();
IAsyncResult^ asyncResult = serviceProxy->BeginProcess(data,
gcnew AsyncCallback(&Processor::ProcessHandler),serviceProxy);
PerformProcessing();
PerformAdditionalProcessing(ret);
```

B. Implement the Web service call as follows:

```
ProcessService^ serviceProxy = gcnew ProcessService();
IAsyncResult^ asyncResult =
serviceProxy->BeginProcess(data,nullptr,nullptr);
PerformProcessing();
while (!asyncResult->IsCompleted){ }String^ ret =
serviceProxy->EndProcess(asyncResult);
PerformAdditionalProcessing(ret);
```

C. Implement the Web service call as follows:

```
ProcessService^ serviceProxy = gcnew ProcessService();
IAsyncResult^ asyncResult =
serviceProxy->BeginProcess(data, nullptr, nullptr);
String^ ret = serviceProxy->EndProcess(asyncResult);
PerformProcessing();
PerformAdditionalProcessing(ret);
```

D. Implement the Web service call as follows:

```
ProcessService^ serviceProxy = gcnew ProcessService();
IAsyncResult^ asyncResult =
serviceProxy->BeginProcess(data, nullptr, nullptr);
PerformProcessing();
String^ ret = serviceProxy->EndProcess(asyncResult);
PerformAdditionalProcessing(ret);
```

Answer: D

QUESTION 74

You are creating a Windows-based application that allows users to store photographs remotely by using a Web service. Both the Web service and the application are configured to use Web Services Enhancements (WSE) 3.0. You need to configure the application to ensure that a photograph can be sent to the Web service. What should you do?

A. Write the following code segment in the application.

```
ImageServiceWse^ serviceProxy = gnew ImageServiceWse();
array<Byte>^ img = GetImageBytes();
serviceProxy->SaveImage(img, imageName);
```

B. Write the following code segment in the application.

```
ImageServiceWse^ serviceProxy = gnew ImageServiceWse();
array<Byte>^ img = GetImageBytes();
serviceProxy->RequestSoapContext->Add(imageName, img);
serviceProxy->SaveImage(nullptr, imageName);
```

C. Write the following code segment in the application.

```
ImageService^ serviceProxy = gnew ImageService();serviceProxy->UserAgent =
"Microsoft WSE 3.0";
array<Byte>^ img = GetImageBytes();
serviceProxy->SaveImage(img, imageName);
```

D. Write the following code segment in the application.

```
ImageService^ serviceProxy = gnew ImageService();
serviceProxy->SoapVersion = SoapProtocolVersion::Soap12;
array<Byte>^ img = GetImageBytes();
serviceProxy->SaveImage(img, imageName);
```

Answer: A

QUESTION 75

A Windows-based application receives messages from a message queue named PriorityQueue. The client application sets the Priority property on the message before sending it. However, received Message objects do not have the Priority property assigned. The following code is used to receive the messages. (Line numbers are included for reference only.)

```
01 MessageQueue^ queue = gnew MessageQueue(".\\PriorityQueue");
02 Message^ m = queue->Receive();
03 Console::WriteLine(m->Priority);
```

You need to ensure that the Windows-based application that receives the messages can access the Priority property. What should you do?

A. Insert the following code segment between lines 01 and 02.

```
DefaultPropertiesToSend^ s = gnew DefaultPropertiesToSend();s->Priority =
MessagePriority::High;queue->DefaultPropertiesToSend = s;
```

B. Insert the following line of code between lines 01 and 02.

```
queue->MessageReadPropertyFilter->Priority = true;
```

C. Replace line 01 with the following code segment.

```
MessageQueue^ queue = gcnew MessageQueue(".\PriorityQueue",  
QueueAccessMode::ReceiveAndAdmin);
```

D. Insert the following code segment between lines 01 and 02.

```
queue->Formatter = gcnew System::Messaging::XmlMessageFormatter(gcnew  
array<Type^> { String::typeid });
```

Answer: B

QUESTION 76

You are writing an application that will run on a portable computer. The application uses a private queue named MyAppQueue to store messages. You need to ensure that the message is retained when the portable computer is restarted. What should you do?

A. Write the following code segment to send the message.

```
MessageQueue^ q = gcnew MessageQueue(".$Private\MyAppQueue");  
Message^ m = gcnew Message("My message body");  
m->UseJournalQueue = true;q->Send(m, "message");
```

B. Write the following code segment to send the message.

```
MessageQueue^ q = gcnew MessageQueue(".$Private\MyAppQueue");  
Message^ m = gcnew Message("My message body");  
m->Recoverable = true;q->Send(m, "message");
```

C. Write the following code segment to send the message.

```
MessageQueue^ q = gcnew MessageQueue(".$Private\MyAppQueue");  
Message^ m = gcnew Message("My message body");  
m->Priority = MessagePriority::High;q->Send(m, "message");
```

D. Write the following code segment to send the message.

```
MessageQueue^ q = gcnew MessageQueue(".$Private\MyAppQueue");  
Message^ m = gcnew Message("My message body");  
MessageQueueTransaction^ trans = gcnew MessageQueueTransaction();  
trans->Begin();q->Send(m, "message", trans);  
trans->Commit();
```

Answer: B

QUESTION 77

When you execute a client application, the following exception is thrown. EncryptedKeyToken is expected but not present in the security header of the incoming message. You discover that the exception is thrown when the client application invokes a Web service named Math with the following code. (Line numbers are included for reference only.)

```
01 try  
02 {  
03
```

```
MathWse^ ws = gcnew MathWse ();
04
int result = ws->Add(1, 2);
05 }
06 catch (Exception^ ex)
07 {
08
MessageBox::Show(ex->Message);
09 }
```

The client application and Web service have the same Web Services Enhancements (WSE) 3.0 policy. The policy configuration file contains the following policy section.

```
<policy name="Secure">
<anonymousForCertificateSecurity
establishSecurityContext="false"
renewExpiredSecurityContext="true"
requireSignatureConfirmation="false"
messageProtectionOrder="SignBeforeEncrypt"
requireDerivedKeys="true"
ttlInSeconds="300">
<!-- XML defining the serviceToken and protection -->
</anonymousForCertificateSecurity>
</policy>
```

You need to ensure that the client application can communicate with the Web service. What should you do?

- A. Add the following code between lines 03 and 04. `ws->SetPolicy("Secure");`
- B. Replace line 03 with the following code. `Math^ ws = gcnew Math();`
- C. Add the following code between lines 03 and 04. `UsernameToken^ u = gcnew UsernameToken("userid", "password"); EncryptedKeyToken^ et = gcnew EncryptedKeyToken(u); ws->SetClientCredential(et);`
- D. Add the following code between lines 03 and 04. `ws->UseDefaultCredentials = true;`
- E. Add the following code between lines 03 and 04. `ws->SetPolicy("anonymousForCertificateSecurity");`

Answer: A

Explanation:

There are two techniques for applying a policy with a client; the first is to call the SetPolicy method of a WSE 3.0-generated proxy class, and the other is to annotate the proxy class with a Policy attribute. So the answer would look like either `ws.SetPolicy("Secure")`, or a class definition that looks like `[Policy(typeof(Secure))] ... public partial class MathServiceWse : WebServicesClientProtocol . . .`. Obviously, answer is (A) calls the SetPolicy method on the proxy and passes in the appropriate security, and none of the other answers are close.

QUESTION 78

A client application calls a Web service named Math. Both the client application and Math are configured with a Web Services Enhancements (WSE) 3.0 policy named Secure to validate anonymous access for certificate security. A Web reference to the Math Web service is added to the client application's project using Microsoft Visual Studio 2005. When the client application is built and executed, a SoapException exception is thrown with the following message. The security requirements are not met because the security header is not included in the incoming message. You need to ensure that the application runs without throwing the SoapException exception. What should you do?

- A. Modify the Math proxy class so that it derives from the following protocol. `System::Web::Services::Protocols::SoapHttpClientProtocol`
- B. Add the following attribute to the Math proxy class definition.
`[Microsoft::Web::Services3::Policy("Secure")]`
- C. Add the following attribute to the Math proxy class definition.
`[Microsoft::Web::Services3::Policy("anonymousForCertificateSecurity")]`
- D. Set the Name property for the WebServiceBindingAttribute attribute on the Math proxy class definition to MathWseSoap.

Answer: B

Explanation:

A WSE 3.0 class inherits from the WebServicesClientProtocol class not the SoapHttpClientProtocol class (A), so this answer is incorrect.

Since the idea is to require the policy to be applied to the proxy class, adding the Policy attribute, and specifying the correct policy name will meet the requirements of the question. In this case (B) identifies the correct policy and is correct, while (C) does not, and is not.

The WebServiceBinding attribute (D) declares a binding (similar to an interface) that defines one or more XML Web Service methods. It has nothing to do with security, and is therefore not the answer.