

[2016-New Free Sharing Of Updated 70-513 VCE And PDF Dumps From GreatExam (81-90)

2016 June Microsoft Official New Released 70-513 Q&As in GreatExam.com! 100% Free Download! 100% Pass Guaranteed! GreatExam 70-513 latest updated braindumps including all new added 70-513 exam questions from exam center which guarantees you can 100% success 70-513 exam in your first try! **Following questions and answers are all new published by Microsoft Official Exam Center: (The full version is in the end of the article!!!)** QUESTION 81 You are modifying a Windows Communication Foundation (WCF) service that issues security tokens. The service is accessible through the named pipe protocol. No endpoints are added in the service code. The configuration file for the service is as follows. (Line numbers are included for reference only.) You need to ensure that new and existing client applications can access the service through HTTP and named pipes. What should you do?

```
01 <<configuration>
02   <system.serviceModel>
03     <services>
04       <service name="Contoso.TokenService">
05         <host>
06           <baseAddresses>
07             <add
08               baseAddress="net.pipe://www.contoso.com/tokenpipe" />
09             </baseAddresses>
10           </host>
11         </service>
12       </services>
13     </system.serviceModel>
14   </configuration>
```

www.greatexam.com

- A. Insert the following at line 05: <endpoint address="<http://www.contoso.com>" binding="wsHttpBinding" contract="Contoso.TokenService" /> B. Insert the following at line 05: <endpoint address="<http://www.contoso.com>" binding="basicHttpBinding" contract="Contoso.TokenService" /> C. Insert the following at line 08: <add baseAddress="<http://www.contoso.com>" /> D. Insert the following at line 08: <add baseAddress="<http://www.contoso.com:8080>" /> Answer: C
- QUESTION 82 A Windows Communication Foundation (WCF) service implements the following contract. (Line numbers are included for reference only.)
- ```
01 [ServiceContract]
02 public interface IDataAccessService
03 {
04 [OperationContract]
05 void PutMessage(string message);
06 ...
07 [OperationContract]
08 [FaultContract(typeof(TimeoutFaultException))]
09 [FaultContract(typeof(FaultException))]
10 string SearchMessages(string search);
11 }
```
- The implementation of the SearchMessages method throws TimeoutFaultException exceptions for database timeouts. The implementation of the SearchMessages method also throws an Exception for any other issue it encounters while processing the request. These exceptions are received on the client side as generic FaultException exceptions. You need to implement the error handling code for SearchMessages and create a new channel on the client only if the channel faults. What should you do?
- A. Catch and handle both TimeoutFaultException and FaultException.  
B. Catch both TimeoutFaultException and FaultException. Create a new channel in both cases.  
C. Catch and handle TimeoutFaultException. Catch FaultException and create a new channel.  
D. Catch and handle FaultException. Catch TimeoutFaultException and create a new channel. Answer: C
- QUESTION 83 You are developing a Windows Communication Foundation (WCF) service. Client applications require reliable sessions to access the service. Users report that they receive ServerTooBusyException errors when their client application tries to establish a session. You need to ensure that the service can handle a minimum of 30 client connection requests. Which ReliableSessionBindingElement property should you configure?
- A. MaxRetryCount  
B. MaxTransferWindowSize  
C. MaxPendingChannels  
D. InactivityTimeout Answer: C
- QUESTION 84 You are creating a Windows Communication Foundation (WCF) service that responds using plain-old XML (POX). You have the following requirements: - You must enable the /catalog.svc Items operation to respond using the POX, JSON, or ATOM formats. - You also must ensure that the same URL is used regardless of the result type. - You must determine the response format by using the Accepts HTTP header. What should you do?
- A. Implement the IChannelInitializer interface in the service class.  
B. Implement the System.Runtime.Serialization.IFormatterConverter interface in the service class.  
C. Set the BodyStyle parameter of the WebGet attribute on the operation to WebMessageBodyStyle.WrappedResponse.  
D. Set the return type of the operation to System.ServiceModel.Channels.Message. Use the current WebOperationContext methods to return the data in the required format. Answer: D
- QUESTION 85 You are developing a Windows Communication Foundation (WCF) service. One of the service operations contains the following code.

```
private static int counter = 0;
[OperationContract]
public void IncrementCount()
{
 counter++;
}
```

You need to set a service behavior that prevents two or more threads from incrementing the counter variable at the same time. Which code segment should you use to set the service behavior?

A. [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single, ConcurrencyMode = ConcurrencyMode.Single)]  
B. [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession, ConcurrencyMode = ConcurrencyMode.Single)]  
C. [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single,

ConcurrencyMode = ConcurrencyMode.Multiple)]D. [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall, ConcurrencyMode = ConcurrencyMode.Reentrant)] Answer: A QUESTION 86 You develop a Windows Communication Foundation (WCF) service that employees use to access bonus information. You define the following service contract. (Line numbers are included for reference only.)

```
01 [ServiceContract(SessionMode = SessionMode.Required)]
02 public interface IFinancialService
03 {
04 [OperationContract]
05 string Login(int employeeID, string passwordHash);
06
07 [OperationContract]
08 double GetBonus(int month);
09
10 [OperationContract(IsTerminating = true)]
11 void Logout();
12 }
```

Client application can invoke methods without logging in. You need to ensure that the client applications invoke Login before invoking any other method. You also need to ensure that client applications cannot consume the service after invoking Logout. Which two action should you perform? (Each correct answer presents part of the solution. Choose two.)

A. Replace line 04 with the following code.  
 [OperationContract(IsInitiating = false)]  
 B. Replace line 04 with the following code.  
 [OperationContract(IsInitiating = true, IsTerminating = true)]  
 C. Replace line 07 with the following code.  
 [OperationContract(IsInitiating = false)]  
 D. Replace line 10 with the following code.  
 [OperationContract(IsInitiating = false, IsTerminating = true)]

Answer: CD

QUESTION 87 Drag and Drop Question You have a client application that consumes a Windows Communication Foundation (WCF) service. The service contains a class named RegistrationService. The RegistrationService class contains two methods named SubmitRegistration and GetRegistrations. The service has the following requirements:- The client application must call the SubmitRegistration method synchronously. - The client application must call the GetRegistrations method asynchronously. You need to complete the client-side code to call the WCF service. Which six code segments should you use in sequence? (To answer, move the appropriate six code segments from the list of code segments to the answer area and arrange them in the correct order.)

|                                                                                                                             | Answer Area |
|-----------------------------------------------------------------------------------------------------------------------------|-------------|
| registration.Surname = "Surname";                                                                                           |             |
| client.GetRegistrationsCompleted += new EventHandler<GetRegistrationsCompletedEventArgs>(client_GetRegistrationsCompleted); |             |
| Registration registration = new Registration();                                                                             |             |
| client.GetRegistrationsCompleted += new EventHandler<GetRegistrationsCompletedEventArgs>(client_GetRegistrationsCompleted); |             |
| client.GetRegistrationsCompleted();                                                                                         |             |
| client.SubmitRegistration(registration);                                                                                    |             |
| client.GetRegistrationsAsync();                                                                                             |             |
| RegistrationServiceClient client = new RegistrationServiceClient("WSHttpBinding_IRegistrationService");                     |             |

Answer:

|                                                                                                                             | Answer Area                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| registration.Surname = "Surname";                                                                                           | RegistrationServiceClient client = new RegistrationServiceClient("WSHttpBinding_IRegistrationService");                     |
| client.GetRegistrationsCompleted += new EventHandler<GetRegistrationsCompletedEventArgs>(client_GetRegistrationsCompleted); | client.GetRegistrationsCompleted += new EventHandler<GetRegistrationsCompletedEventArgs>(client_GetRegistrationsCompleted); |
| Registration registration = new Registration();                                                                             | client.GetRegistrationsAsync();                                                                                             |
| client.GetRegistrationsCompleted += new EventHandler<GetRegistrationsCompletedEventArgs>(client_GetRegistrationsCompleted); | Registration registration = new Registration();                                                                             |
| client.GetRegistrationsCompleted();                                                                                         | registration.Surname = "Surname";                                                                                           |
| client.SubmitRegistration(registration);                                                                                    | client.SubmitRegistration(registration);                                                                                    |
| client.GetRegistrationsAsync();                                                                                             |                                                                                                                             |
| RegistrationServiceClient client = new RegistrationServiceClient("WSHttpBinding_IRegistrationService");                     |                                                                                                                             |

www.greatexam.com

QUESTION 88 You are creating a Windows Communication Foundation (WCF) service based on WSHttpBinding. New audit requirements dictate that callers must be authenticated on every call to ensure that their credentials have not been revoked. You need to ensure that the service will not cache the security request token. What should you do?

A. Apply a ServiceBehavior attribute to the service implementation class with the InstanceContextMode property set to Single.  
 B. In the message security configuration, change clientCredentialType from IssuedToken to UserName.  
 C. In the message security configuration, set

establishSecurityContext to false.D. At the end of every operation, call the SessionStateUtility.RaiseSessionEnd method. Answer: C QUESTION 89 You develop a Windows Communication Foundation (WCF) service to generate reports. Client applications call the service to initiate report generation but do not wait for the reports to be generated. The service does not provide any status to the client applications. The service class is defined as follows. (Line numbers are included for reference only.) You need to ensure that client applications can initiate reports without waiting for status. Which two actions should you perform (Each correct answer presents part of the solution. Choose two.)

```

01 [ServiceContract]
02 public class ReportGeneratorService
03 {
04
05 private int GenerateReports(int clientID)
06 {
07 ...
08 return 0;
09 }
10 }

```

www.greatexam.com

A. Insert the following code at line 04. [OperationContract(IsOneWay=true)] B. Insert the following code at line 04. [OperationContract(AsyncPattern=false)] C. At line 05, change the GenerateReports method from private to public. D. Remove line 08. At line 05, change the return type of GenerateReports method to void. Answer: AD QUESTION 90 You are developing a Windows Communication Foundation (WCF) service. The following code defines and implements the service. (Line numbers are included for reference only.) You need to ensure that two identical copies of the received message are created in the service. Which code segment should you insert at line 12?

```

01 [ServiceContract(SessionMode = SessionMode.Allowed)]
02 public interface ICatchAll
03 {
04 [OperationContract(IsOneWay = false, Action = "*",
05 ReplyAction = "*")]
06 Message ProcessMessage(Message message);
07 }
08 public class CatchAllService : ICatchAll
09 {
10 public Message ProcessMessage(Message message)
11 {
12
13 ...
14 return returnMsg;
15 }
16 }

```

www.greatexam.com

A. Message msgCopy = message.CreateBufferedCopy(8192) as Message; Message returnMsg = message.CreateBufferedCopy(8192) as Message; B. MessageBuffer buffer = message.CreateBufferedCopy(8192); Message msgCopy = buffer.CreateMessage(); Message returnMsg = buffer.CreateMessage(); C. MessageBuffer buffer = message.CreateBufferedCopy(8192); Message msgCopy = buffer.CreateMessage(); Message returnMsg = msgCopy; D. Message msgCopy = message; Message returnMsg = message; Answer: B GreatExam regular updates of Microsoft 70-513 preparation materials, with accurate answers, keeps the members one step ahead in the real 70-513 exam. The experts with more than 10 years experience in Certification Field work with us. 2016 Microsoft 70-513 exam dumps (All 341 Q&As) from GreatExam:

<http://www.greatexam.com/70-513-exam-questions.html> [100% Exam Pass Guaranteed!!!]