QUESTION 1

You are creating a Windows Communication Foundation (WCF) service that is implemented as follows. (Line numbers are included for reference only.)

```
01 [ServiceContract]
02 [ServiceBehavior(IncludeExceptionDetailsInFaults = true)] 03 public
class OrderService
04 {
05 [OperationContract]
06 public void SubmitOrder(Order anOrder)
07 {
08 try
09 {
10 ...
11 }
12 catch(DivideByZeroException ex)
13 {
14 ...
15 }
16 }
17 }
```

You need to ensure that the stack trace details of the exception are not included in the error information sent to the client. What should you do?

A. Replace line 14 with the following line: throw;
B. Replace line 14 with the following line:
   throw new FaultException<Order>(anOrder, ex.ToString());
C. After line 05, add the following line:
   [FaultContract(typeof(FaultException<Order>))]
   Replace line 14 with the following line: throw ex;
D. Alter line 05, add the following line:
   [FaultContract(typeof(FaultException<Order>))]
   Replace line 14 with the following line:
   throw new FaultException<Order>(anOrder, "Divide by zero exception");

Answer: D

QUESTION 2

You are creating a Windows Communication Foundation (WCF) service. You do not want to expose the internal implementation at the service layer.
You need to expose the following class as a service named Arithmetic with an operation named Sum:

```
public class Calculator
public int Add(int x, int y)
```

Which code segment should you use?

A. [ServiceContract(Namespace="Arithmetic")]
   public class Calculator
   [Operation Contract(Action="Sum")]
   public int Add(int x, int y)
B. [ServiceContract(ConfigurationName="Arith

metic")]
public class Calculator
[Operation Contract(Action="Sum")]
public int Add(int x, int y)

C. [ServiceContract(Name="Arithmetic")]
public class Calculator
[OperationContract(Name="Sum")]
public int Add(int x, int y)

D. [ServiceContract(Name="Arithmetic")]
public class Calculator
[OperationContract(ReplyAction="Sum")]
public int Add(int x, int y)

Answer: C

QUESTION 3
You are developing a data contract for a Windows Communication Foundation (WCF) service.
The data in the data contract must participate in round trips. Strict schema validity is not required.
You need to ensure that the contract is forward-compatible and allows new data members to be
added to it. Which interface should you implement in the data contract class?

A. ICommunicationObject
B. IExtension<T>
C. IExtensibleObject<T>
D. IExtensibleDataObject

Answer: D

QUESTION 4
A Windows Communication Foundation (WCF) application uses a data contract that has several
data members.
You need the application to throw a SerializationException if any of the data members are not
present when a serialized instance of the data contract is deserialized. What should you do?

A. Add the KnownType attribute to the data contract.
   Set a default value in each of the data member declarations.
B. Add the KnownType attribute to the data contract.
   Set the Order property of each data member to unique integer value.
C. Set the EmitDefaultValue property of each data member to false.
D. Set the IsRequired property of each data member to true.

Answer: D

QUESTION 5
A Windows Communication Foundation (WCF) application uses the following data contract

```
[DataContract]
public class Person
[DataMember]
public string firstName;
[DataMember]
```

```
public string lastName;
[DataMember]
public int age;
[DataMember]
public int ID;
```

You need to ensure that the following XML segment is generated when the data contract is serialized.

```
<Person>
<firstName xsi:nil="true"/>
<lastName xsi:nil="true"/>
<ID>999999999<ID>
</Person>
```

Which code segment should you use?

A. [DataMember]
   public string firstName;
   [DataMember]
   public string lastName;
   [DataMember(EmitDefaultValue = true)]
   public int age = 0;
   [DataMember(EmitDefaultvValue = true)]
   public int ID = 999999999;
B. [DataMember(EmitDefaultValue = false)]
   public string firstName = null;
   [DataMember(EmitDefaultValue = false)]
   public string lastName = null;
   [DataMember(EmitDefaultValue = true)]
   public int age = -1;
   [DataMember(EmitDefaultValue = false)]
   public int ID = 999999999;
C. [DataMember(EmitDefaultValue = true)]
   public string firstName;
   [DataMember(EmitDefaultValue = true)]
   public string lastName;
   [DataMember(EmitDefaultValue = false)]
   public int age = -1;
   [DataMember(EmitDefaultValue = false)]
   public int ID = 999999999;
D. [DataMember]
   public string firstName = null;
   [DataMember]
   public string lastName = null;
   [DataMember(EmitDefaultValue = false)]
   public int age = 0;
   [DataMember(EmitDefaultValue = false)]
   public int ID = 999999999;

Answer: D

QUESTION 6
The following is an example of a SOAP envelope.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope">
<s:Header>
<h:StoreId xmlns:h="http://www.contoso.com">6495</h:StoreId>
</s:Header>
<s:Body>
<CheckStockRequest xmlns="http://www.contoso.com">
<ItemId>2469<ItemId>
</CheckStockRequest>
</s: Body>
</s:Envelope>
```

You need to create a message contract that generates the SOAP envelope.
Which code segment should you use?

A. [MessageContract(WrapperName="http://www.contoso.com")]
   public class CheckStockRequest
   [MessageHeader(Namespace="http://www.contoso.com")]
   public int StoreId { get; set; }
   [MessageBodyMember(Namespace="http://www.contoso.com")]
   public int ItemId { get; set; }
B. [MessageContract(WrapperNamespace="http://www.contoso.com")] public class
   CheckStockRequest
   [MessageHeader(Namespace="http://www.contoso.com")]
   public int StoreId { get; set; }
   [MessageBodyMember(Namespace="http://www contoso.com")]
   public int ItemId { get; set; }
C. [MessageContract(WrapperNamespace="http://www.contoso.com")] public class
   CheckStockRequest
   [MessageHeader(Namespace="http://www.contoso.com")]
   public int StoreId { get; set; }
   public int ItemId { get; set; }
D. [MessageContract(WrapperNamespace="http://www.contoso.com")] public class
   CheckStockRequest
   [MessageHeader(Namespace="http://www.contoso.com")]
   public int StoreId { get; set; }
   [MessageBodyMember]
   public int ItemId { get; set; }

Answer: D

QUESTION 7
You are developing a client that sends several types of SOAP messages to a Windows
Communication Foundation (WCF) service method named PostData. PostData is currently
defined as follows:

```
[OperationContract] void PostData(Order data);
```

You need to modify PostData so that it can receive any SOAP message. Which code segment
should you use?

A. [OperationContract(IsOneWay=true, Action="*", ReplyAction="*")] void PostData(Order data);
B. [OperationContract(IsOneWay=true, Action="*", ReplyAction = "*")] void PostData(BodyWriter
   data);

C. [OperationContract]
   void PostData(BodyWriter data);
D. [OperationContract]
   void PostData(Message data);

Answer: D

QUESTION 8
A class named TestService implements the following interface:

```
[ServiceContract]
public interface ITestService
[OperationContract]
DateTime GetServiceTime();
```

TestService is hosted in an ASP.NET application.
You need to modify the application to allow the GetServiceTime method to return the data formatted as JSON.
It must do this only when the request URL ends in /ServiceTime. What should you do?

A. Add this attribute to the GetServiceTime method.
   [WebInvoke(Method="POST")]
   In the web.config file, add this element to system.serviceModel/behaviors/endpointBehaviors.
   <behavior name="Json">
   </behavior>
   In the web.config file, configure TestService in the system.serviceModel/services collection as follows:
   <service name="TestService">
   <endpoint address="/ServiceTime"
   contract="TestService"
   behaviorConfiguration="Json"
   binding="webHttpBinding" />
   </service>
B. Add this attribute to the GetServiceTime method.
   [WebInvoke(Method="GET", UriTemplate="/ServiceTime",
   ResponseFormat=WebMessageFormat.Json)]
   In the web.config file, configure TestService in the system.serviceModel/services collection as follows:
   <service name="TestService">
   <endpoint address="/ServiceTime"
   contract="TestService"
   binding="webHttpBinding"/>
   </service>
C. Add this attribute to the GetServiceTime method
   [WebGet(ResponseFormat=WebMessageFormat.Json, UriTemplate="/ServiceTime")] Create
   a new svc file named Jsonversion.svc with the following content.
   <% @ServiceHost Service="TestService" Factory="System.ServiceModel.
   ActivationWebServiceHostFactory" %>
D. Add this attribute to the GetServiceTime method.
   [WebGet(UriTemplate="Json)/ServiceTime")]
   Create a new .svc file named Jsonversion.svc with the following content <% @ServiceHost
   Service="TestService" Factory="System.ServiceModel.
   ActivationWebServiceHostFactory" %>

Answer: C

QUESTION 9
You are creating a Windows Communication Foundation (WCF) service that implements operations in a RESTful manner.
You need to add a delete operation. You implement the delete method as follows:

```
void DeleteItems(string id);
```

You need to configure WCF to call this method when the client calls the service with the HTTP DELETE operation. What should you do?

A. Add the WebInvoke(UriTemplate="/Items/{id}", Method="DELETE") attribute to the operation
B. Add the HttpDelete atribute to the operation
C. Replace the string parameter with a RemovedActivityAction parameter
D. Replace the return type with RemovedActivityAction.

Answer: A

QUESTION 10
A Windows Communication Foundation (WCF) service uses the following service contract.

```
[ServiceContract]
public interface IService
[OperationContract]
string Operation1(string s);
```

You need to ensure that the operation contract Operation1 responds to HTTP POST requests.
Which code segment should you use?

A. [OperationContract]
   [WebInvoke(Method="POST")]
   string Operation1(string s);
B. [OperationContract]
   [WebGet(UriTemplate="POST")]
   string Operation1(string s);
C. [OperationContract(ReplyAction="POST")]
   string Operation1(string s);
D. [OperationContract(Action="POST")]
   string Operation1(string s);

Answer: A

Thank You for Trying Our Product

Lead2pass Certification Exam Features:

★ More than 99,900 Satisfied Customers Worldwide.

★ Average 99.9% Success Rate.

★ Free Update to match latest and real exam scenarios.

★ Instant Download Access! No Setup required.

★ Questions & Answers are downloadable in PDF format and VCE test engine format.

★ Multi-Platform capabilities - Windows, Laptop, Mac, Android, iPhone, iPod, iPad.

★ 100% Guaranteed Success or 100% Money Back Guarantee.

★ Fast, helpful support 24x7.

View list of all certification exams:

http://www.lead2pass.com/all-products.html

10% Discount Coupon Code: ASTR14