I S L E V E R

70-513

TS: Windows Communication Foundation
Development with Microsoft .NET Framework 4

DEMO

https://www.islever.com/70-513.html
https://www.islever.com/microsoft.html

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 {
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. After 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
  {
    [OperationContract(Action="Sum")]
    public int Add(int x, int y)
    {
    ...
    }
  }
B.[ServiceContract(ConfigurationName="Arithmetic")]
  public class Calculator
  {
    [OperationContract(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
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.