

KTest

更に上のクオリティ 更に上のサービス



問題集

<http://www.ktest.jp>

1年で無料進級することに提供する

Exam : **70-569**

Title : Upgrade: Transition your
MCPD Enterprise
Application Developer Skills
to MCPD Enterprise
Applications Developer 3.5,
Part 2

Version : Demo

1. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment.

```
namespace MyServices
{
    [ServiceContract()]
    interface IManageOrders
    {
        ...
    }
}
```

The service metadata must be exposed at the relative address named meta.

You need to add an endpoint element to the app.config file of the service host.

Which code fragment should you add

- A. `<endpoint address="meta" binding="wsHttpBinding" contract="IManageOrders" />`
- B. `<endpoint address="meta" binding="wsHttpBinding" contract="MyServices.IMetadataExchange" />`
- C. `<endpoint address="meta" binding="mexHttpBinding" contract="IMetadataExchange" />`
- D. `<endpoint address="meta" binding="mexHttpBinding" contract="MyServices.IManageOrders" />`

Answer: C

2. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment.

```
Namespace MyServices
    <ServiceContract()>Interface IManageOrders
    ...
```

End Interface

End Namespace

The service metadata must be exposed at the relative address named meta.

You need to add an endpoint element to the app.config file of the service host.

Which code fragment should you add

- A. `<endpoint address="meta" binding="wsHttpBinding" _
contract="IMangeOrders" />`
- B. `<endpoint address="meta" binding="wsHttpBinding" _
contract="MyServices.IMetadadataExchange" />`
- C. `<endpoint address="meta" binding="mexHttpBinding" _
contract="IMetadadataExchange" />`
- D. `<endpoint address="meta" binding="mexHttpBinding" _
contract="MyServices.IMangeOrders" />`

Answer: C

3. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework

3.5. You have successfully defined a service contract named IMangeOrders.

You write the following code segment.

```
public class OrderImpl : IMangeOrders {  
    public void MarkOrderClosed(int orderId){  
        try {  
            ...  
        }  
        catch (SqlException exc){  
            throw new FaultException<DataFault>(new DataFault());  
        }  
    }  
}
```

[DataContract]

```
public class DataFault {
```

```
}
```

You need to create a fault contract for the MarkOrderClosed method on the IManageOrders service contract.

Which code segment should you add

- A. [FaultContract(typeof(DataFault))]
- B. [FaultContract(typeof(Exception))]
- C. [FaultContract(typeof(SqlException))]
- D. [FaultContract(typeof(FaultException))]

Answer: A

4. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework

3.5. You have successfully defined a service contract named IManageOrders.

You write the following code segment.

```
Public Class OrderImpl
```

```
Implements IManageOrders
```

```
Public Sub MarkOrderClosed(ByVal orderId As Integer) _
```

```
Implements IManageOrders.MarkOrderClosed
```

```
Try
```

```
...
```

```
Catch ex As SqlException
```

```
Throw New FaultException(Of DataFault)( _
```

```
New DataFault())
```

```
End Try
```

```
End Sub
```

```
End Class
```

```
<DataContract()> _
```

```
Public Class DataFault
```

```
End Class
```

You need to create a fault contract for the MarkOrderClosed method on the IManageOrders service contract.

Which code segment should you add

- A. <FaultContract(GetType(DataFault))>
- B. <FaultContract(GetType(Exception))>
- C. <FaultContract(GetType(SqlException))>
- D. <FaultContract(GetType(FaultException))>

Answer: A

5. You have created a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

The existing service interface is named IMyService, and contains the following code segment.

```
[ServiceContract(Name="SvcOrder",  
    Namespace="http://contoso.com/services")]
```

```
public interface IMyService
```

```
{
```

```
    [OperationContract]
```

```
    void DoSomething();
```

```
}
```

You create a new service named IMyServiceV1 that contains an operation named DoSomethingElse.

You need to ensure that existing client applications are still able to access the IMyService.DoSomething method without modifying client code.

Which code segment should you use

A. [ServiceContract(Namespace="http: //contoso.com/services/V1")]

```
public interface IMyServiceV1 : IMyService
```

```
{
```

```
    [OperationContract]
```

```
    void DoSomethingElse();
```

```
}
```

B. [ServiceContract(Name="SvcOrder")]

```
public interface IMyServiceV1 : IMyService
```

```
{
```

```
[OperationContract]
void DoSomethingElse();
}
C. [ServiceContract(Name="SvcOrderV1",
Namespace="http://contoso.com/services")]
public interface IMyServiceV1 : IMyService
{
[OperationContract]
void DoSomethingElse();
}
D. [ServiceContract(Name="SvcOrder",
Namespace="http://contoso.com/services")]
public interface IMyServiceV1 : IMyService
{
[OperationContract]
void DoSomethingElse();
}
```

Answer: D

6. You have created a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

The existing service interface is named IMyService, and contains the following code segment.

```
<ServiceContract(Name:="SvcOrder", _
Namespace:="http://contoso.com/services")> _
Public Interface IMyService
    <OperationContract()> _
    Sub DoSomething()
End Interface
```

You create a new service named IMyServiceV1 that contains an operation named DoSomethingElse.

You need to ensure that existing client applications are still able to access the IMyService.DoSomething

method without modifying client code.

Which code segment should you use

A. <ServiceContract(Namespace:="http: //contoso.com/services/V1")> _

Public Interface IMyServiceV1

Inherits IMyService

<OperationContract()> _

Sub DoSomethingElse()

End Interface

B. <ServiceContract(Name:="SvcOrder")> _

Public Interface IMyServiceV1

Inherits IMyService

<OperationContract()> _

Sub DoSomethingElse()

End Interface

C. <ServiceContract(Name:="SvcOrderV1", _

Namespace:="http: //contoso.com/services")> _

Public Interface IMyServiceV1

Inherits IMyService

<OperationContract()> _

Sub DoSomethingElse()

End Interface

D. <ServiceContract(Name:="SvcOrder", _

Namespace:="http: //contoso.com/services")> _

Public Interface IMyServiceV1

Inherits IMyService

<OperationContract()> _

Sub DoSomethingElse()

End Interface

Answer: D

7. You create a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

The service contains the following code segment.

```
[DataContract]
public class Person
{
    &

}
```

```
[DataContract]
public class Customer : Person
{
    &

}
```

You need to create a service contract that meets the following requirements:

The service contract must have an operation contract named GetPerson that returns an object of type Person.

The GetPerson operation must be able to return an object of type Customer.

Which code segment should you use

A. [ServiceContract]

```
[ServiceKnownType("GetPerson")]
public interface IMyService
{
    [OperationContract]
    Person GetPerson();
}
```

B. [ServiceContract]

```
public interface IMyService
{
    [OperationContract]
```

```
[ServiceKnownType("Customer")]
    Person GetPerson();
}
```

C. [ServiceContract]

```
[ServiceKnownType(typeof(Customer))]
public interface IMyService
{
    [OperationContract]
    Person GetPerson();
}
```

D. [ServiceContract]

```
[ServiceKnownType("GetPerson",typeof(Customer))]
public interface IMyService
{
    [OperationContract]
    Person GetPerson();
}
```

Answer: C

8. You create a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

The service contains the following code segment.

```
<DataContract(> _
```

```
Public Class Person
```

```
...
```

```
End Class
```

```
<DataContract(> _
```

```
Public Class Customer
```

```
    Inherits Person
```

```
...
```

```
End Class
```

You need to create a service contract that meets the following requirements:

The service contract must have an operation contract named GetPerson that returns an object of type Person.

The GetPerson operation must be able to return an object of type Customer.

Which code segment should you use

A. <ServiceContract()> _

<ServiceKnownType("GetPerson")> _

Public Interface IMyService

<OperationContract()> _

Function GetPerson() As Person

End Interface

B. <ServiceContract()> _

Public Interface IMyService

<OperationContract()> _

<ServiceKnownType("Customer")> _

Function GetPerson() As Person

End Interface

C. <ServiceContract()> _

<ServiceKnownType(GetType(Customer))> _

Public Interface IMyService

<OperationContract()> _

Function GetPerson() As Person

End Interface

D. <ServiceContract()> _

<ServiceKnownType("GetPerson", GetType(Customer))> _

Public Interface IMyService

<OperationContract()> _

Function GetPerson() As Person

End Interface

Answer: C

9. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework

3.5. The service will contain an enumeration named OrderState.

The OrderState enumeration will contain the following four values:

Processing

Cancelled

Confirmed

Closed

The client application must be able to set the state of an Order entity to only the following two values:

Cancelled

Closed

You need to create the data contract for OrderState.

Which code segment should you use

A. [DataContract]

```
public enum OrderState
```

```
{  
    Processing=1,  
    [DataMember]  
    Cancelled=2,  
    [DataMember]  
    Confirmed=3,  
    Closed=4
```

```
}
```

B. [DataContract]

```
public enum OrderState
```

```
{  
    Processing=1,  
    [EnumMember]  
    Cancelled=2,  
    Confirmed=3,
```

```
[EnumMember]
    Closed=4
}
```

C. [DataContract]

```
public enum OrderState
{
    [EnumMember(Value="False")]
    Processing=1,
    [EnumMember(Value="True")]
    Cancelled=2,
    [EnumMember(Value="True")]
    Confirmed=3,
    [EnumMember(Value="False")]
    Closed=4
}
```

D. [DataContract]

```
public enum OrderState
{
    [DataMember]
    Processing=1,
    [DataMember(IsRequired=true)]
    Cancelled=2,
    [DataMember]
    Confirmed=3,
    [DataMember(IsRequired=true)]
    Closed=4
}
```

Answer: B

10. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework

3.5. The service will contain an enumeration named OrderState.

The OrderState enumeration will contain the following four values:

Processing

Cancelled

Confirmed

Closed

The client application must be able to set the state of an Order entity to only the following two values:

Cancelled

Closed

You need to create the data contract for OrderState.

Which code segment should you use

A. <DataContract()> _

```
Public Enum OrderState
```

```
    Processing = 1
```

```
    <DataMember()> _
```

```
    Cancelled = 2
```

```
    <DataMember()> _
```

```
    Confirmed = 3
```

```
    Closed = 4
```

```
End Enum
```

B. <DataContract()> _

```
Public Enum OrderState
```

```
    Processing = 1
```

```
    <EnumMember()> _
```

```
    Cancelled = 2
```

```
    Confirmed = 3
```

```
    <EnumMember()> _
```

```
    Closed = 4
```

```
End Enum
```

C. <DataContract()> _

Public Enum OrderState

<EnumMember(Value:="False")> _

Processing = 1

<EnumMember(Value:="True")> _

Cancelled = 2

<EnumMember(Value:="True")> _

Confirmed = 3

<EnumMember(Value:="False")> _

Closed = 4

End Enum

D. <DataContract()> _

Public Enum OrderState

<DataMember()> _

Processing = 1

<DataMember(IsRequired:=True)> _

Cancelled = 2

<DataMember()> _

Confirmed = 3

<DataMember(IsRequired:=True)> _

Closed = 4

End Enum

Answer: B

11. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You need to ensure that the service can send data in the following format to the client applications.

<Account Id="">

<Name> </Name>

<Balance Currency=""> </Balance>

</Account>

Which code segment should you use

A. [Serializable]

```
public class Account
{
    [XmlAttribute]
    public string Id;
    [XmlElement]
    public string Name;
    [XmlAttribute]
    public string Currency;
    [XmlElement]
    public double Balance;
}
```

B. [DataContract]

```
public class Account
{
    [DataMember(Order=0)]
    public string Id;
    [DataMember(Order=1)]
    public string Name;
    [DataMember(Order=0)]
    public double Balance;
    [DataMember(Order=1)]
    public string Currency;
}
```

C. [Serializable]

```
public class Account
{
    [XmlAttribute]
    public string Id;
```



```
public string Name;  
[XmlElement("Balance")]  
public BalanceVal Balance;  
}
```

```
[Serializable]
```

```
public class BalanceVal
```

```
{  
    [XmlText]  
    public double Amount;  
    [XmlAttribute]  
    public string Currency;  
}
```

```
D. [DataContract]
```

```
public class Account
```

```
{  
    [DataMember(Order=0)]  
    public string Id;  
    [DataMember(Order=1)]  
    public string Name;  
    [DataMember(Name="Balance", Order=2)]  
    public BalanceVal Balance;  
}
```

```
[DataContract]
```

```
public struct BalanceVal
```

```
{  
    [DataMember(Order=0)]  
    public double Balance;  
    [DataMember(Order=1)]  
    public string Currency;  
}
```

Answer: C

12. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You need to ensure that the service can send data in the following format to the client applications.

```
<Account Id="">  
  <Name> </Name>  
  <Balance Currency=""> </Balance>  
</Account>
```

Which code segment should you use

A. <Serializable()> _

```
Public Class Account
```

```
  <XmlAttribute()> _
```

```
  Public Id As String
```

```
  <XmlElement()> _
```

```
  Public Name As String
```

```
  <XmlAttribute()> _
```

```
  Public Currency As String
```

```
  <XmlElement()> _
```

```
  Public Balance As Double
```

```
End Class
```

B. <DataContract()> _

```
Public Class Account
```

```
  <DataMember(Order:=0)> _
```

```
  Public Id As String
```

```
  <DataMember(Order:=1)> _
```

```
  Public Name As String
```

```
  <DataMember(Order:=0)> _
```

```
  Public Balance As Double
```

```
  <DataMember(Order:=1)> _
```

Public Currency As String

End Class

C. <Serializable()> _

Public Class Account

<XmlAttribute()> _

Public Id As String

Public Name As String

<XmlElement("Balance")> _

Public Balance As BalanceVal

End Class

<Serializable()> _

Public Class BalanceVal

<XmlText()> _

Public Amount As Double

<XmlAttribute()> _

Public Currency As String

End Class

D. <DataContract()> _

Public Class Account

<DataMember(Order:=0)> _

Public Id As String

<DataMember(Order:=1)> _

Public Name As String

<DataMember(Name:="Balance", Order:=2)> _

Public Balance As BalanceVal

End Class

<DataContract()> _

Public Structure BalanceVal

<DataMember(Order:=0)> _

Public Amount As Double

```
<DataMember(Order:=1)> _
```

```
Public Currency As String
```

```
End Structure
```

Answer: C

13. You create a Windows Communication Foundation (WCF) service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

```
01 public interface IMyService
02 {
03
04     string ProcessString(string name);
05 }
```

You create a host for the WCF service. You also create a service endpoint at <http://localhost:8080/service>.

You add an instance of the `HttpTransferEndPointBehavior` class to the host.

You need to ensure that the `ProcessString` method can be invoked from a Web browser by using the URL

<http://localhost:8080/service/process name=value>

Which code segment should you insert at line 03

A. `[OperationContract(Name="process", Action="Get")]`

B. `[OperationContract(Name="process", Action="Post")]`

C. `[OperationContract]`

`[WebGet(UriTemplate = "process name={name}")]`

D. `[OperationContract]`

`[WebInvoke(UriTemplate = "process name={name}")]`

Answer: C

14. You create a Windows Communication Foundation (WCF) service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

```
01 Public Interface IMyService
```

02

03 Function ProcessString(ByVal name As String) As String

04 End Interface

You create a host for the WCF service. You also create a service endpoint at `http://localhost:8080/service`.

You add an instance of the `HttpTransferEndPointBehavior` class to the host.

You need to ensure that the `ProcessString` method can be invoked from a Web browser by using the URL

`http://localhost:8080/service/process name=value`

Which code segment should you insert at line 02

A. `<OperationContract(Name:="process", Action:="Get")> _`

B. `<OperationContract(Name:="process", Action:="Post")> _`

C. `<OperationContract()> _`

`<HttpTransferContract(Path:="process", Method:="Get")> _`

D. `<OperationContract()> _`

`<HttpTransferContract(Path:="process", Method:="Post")> _`

Answer: C

15. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

01 `[ServiceContract(Namespace="http://uri.contoso.com")]`

02 `public interface IMyService`

03 `{`

04 `[OperationContract]`

05 `string ProcessDetails(string s);`

06 `[OperationContract(Action="UpdateStatus")]`

07 `void UpdateStatus();`

08

09 `}`

If the existing operation contract is unable to process a request made to the service, a generic operation contract must attempt to process the request.

You need to create the generic operation contract.

Which code segment should you insert at line 08

A. [OperationContract(Action="**")]

```
void ProcessOthers(Message msg);
```

B. [OperationContract(Action="**")]

```
void ProcessOthers();
```

C. [OperationContract(Action="Default")]

```
void ProcessOthers(Message msg);
```

D. [OperationContract(Action="Default")]

```
void ProcessOthers();
```

Answer: A

16. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

```
01 <ServiceContract(Namespace:="http://uri.contoso.com")> _
```

```
02 Public Interface IMyService
```

```
03     <OperationContract> _
```

```
04     Function ProcessDetails(ByVal s As String) As String
```

```
05     <OperationContract(Action:="UpdateStatus")> _
```

```
06     Sub UpdateStatus()
```

```
07
```

```
08 End Interface
```

If the existing operation contract is unable to process a request made to the service, a generic operation contract must attempt to process the request.

You need to create the generic operation contract.

Which code segment should you insert at line 07

A. <OperationContract(Action:="**")> _

```
Sub ProcessOthers(ByVal msg As Message)
```

B. <OperationContract(Action:="**")> _

Sub ProcessOthers()

C. <OperationContract(Action:="Default")> _

Sub ProcessOthers(ByVal msg As Message)

D. <OperationContract(Action:="Default")> _

Sub ProcessOthers()

Answer: A

17. You are creating a Windows Communication Foundation (WCF) service by using Microsoft .NET Framework 3.5.

You configure a binding to enable streaming.

You need to ensure that the client application is able to stream large XML files to the WCF service.

Which operation contract should you create

A. [OperationContract]

void UploadFile(Stream xmlData);

B. [OperationContract]

void UploadFile(XmlWriter xmlData);

C. [OperationContract]

void UploadFile(StreamWriter xmlData);

D. [OperationContract]

void UploadFile(byte[] xmlData);

Answer: A

18. You are creating a Windows Communication Foundation (WCF) service by using Microsoft .NET Framework 3.5.

You configure a binding to enable streaming.

You need to ensure that the client application is able to stream large XML files to the WCF service.

Which operation contract should you create

A. <OperationContract()> _

Sub UploadFile(ByVal xmlData As Stream)

B. <OperationContract()> _

Sub UploadFile(ByVal xmlData As XmlWriter)

C. <OperationContract(> _

Sub UploadFile(ByVal xmlData As StreamWriter)

D. <OperationContract(> _

Sub UploadFile(ByVal xmlData As Byte())

Answer: A

19. You create a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

```
01 [ServiceContract(SessionMode=SessionMode.Required)]
```

```
02 public interface IOrderManager
```

```
03 {
```

```
04
```

```
05     void CloseOrder();
```

```
06 }
```

You need to decorate the operation as the method that closes the current session.

Which code segment should you insert at line 04

A. [OperationContract(IsInitiating=false)]

B. [OperationContract(IsTerminating=true)]

C. [OperationContract]

[OperationBehavior(ReleaseInstanceMode=

ReleaseInstanceMode.AfterCall)]

D. [OperationContract(IsTerminating=false)]

[OperationBehavior(ReleaseInstanceMode=

ReleaseInstanceMode.AfterCall)]

Answer: B

20. You create a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

```
01 <ServiceContract(SessionMode:=SessionMode.Required)> _
```


02 Public Interface IOrderManager

03

04 Sub CloseOrder()

05 End Interface

You need to decorate the operation as the method that closes the current session.

Which code segment should you insert at line 03

A. <OperationContract(IsInitiating:=False)> _

B. <OperationContract(IsTerminating:=True)> _

C. <OperationContract()> _

<OperationBehavior(_

ReleaseInstanceMode:=ReleaseInstanceMode.AfterCall)> _

D. <OperationContract(IsTerminating:=False)> _

<OperationBehavior(_

ReleaseInstanceMode:=ReleaseInstanceMode.AfterCall)> _

Answer: B

21. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You create the following service contract. (Line numbers are included for reference only.)

01 [ServiceContract]

02 public interface IMyService

03 {

04 [OperationContract]

05

06 List<string> GetData(int index);

07 }

You need to ensure that the GetData operation can be accessed by using the URI in the following manner:

GetData/1 for the index value 1

GetData/2 for the index value 2

.

.

GetData/n for the index value n

Which code segment should you insert at line 05

- A. [WebGet(UriTemplate="GetData/{index}",
BodyStyle=WebMessageBodyStyle.Bare)]
- B. [WebGet(UriTemplate="GetData index={index}",
BodyStyle=WebMessageBodyStyle.Bare)]
- C. [WebGet(UriTemplate="GetData/[index]",
BodyStyle=WebMessageBodyStyle.Bare)]
- D. [WebGet[UriTemplate="GetData index={index}",
BodyStyle=WebMessageBodyStyle.Bare]]

Answer: A

22. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You create the following service contract. (Line numbers are included for reference only.)

```
01 <ServiceContract()> _  
02 Public Interface IMyService  
03     <OperationContract()> _  
04  
05     Function GetData(ByVal index As Integer) As List(Of String)  
06 End Interface
```

You need to ensure that the GetData operation can be accessed by using the URI in the following manner:

GetData/1 for the index value 1

GetData/2 for the index value 2

.

.

GetData/n for the index value n

Which code segment should you insert at line 04

- A. <WebGet(UriTemplate:="GetData/{index}", _
BodyStyle:=WebMessageBodyStyle.Bare)> _
- B. <WebGet(UriTemplate:="GetData index={index}", _
BodyStyle:=WebMessageBodyStyle.Bare)> _
- C. <WebGet(UriTemplate:="GetData/[index]", _
BodyStyle:=WebMessageBodyStyle.Bare)> _
- D. <WebGet(UriTemplate:="GetData index=[index]", _
BodyStyle:=WebMessageBodyStyle.Bare)> _

Answer: A

23. You create a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You create the following service contract.

```
[ServiceContract]
```

```
public interface IMath
```

```
{
```

```
    [OperationContract]
```

```
    int Add(int num1, int num2);
```

```
}
```

You need to add an operation contract to perform the Add operation asynchronously.

Which operation contract should you use

A. [OperationContract(AsyncPattern=true)]

```
IAsyncResult BeginAdd(int num1, int num2);
```

```
int EndAdd(IAsyncResult res);
```

B. [OperationContract]

```
int BeginAdd(int num1, int num2, AsyncCallback cb, object state);
```

```
IAsyncResult EndAdd();
```

C. [OperationContract]

```
IAsyncResult BeginAdd(int num1, int num2);
```

```
[OperationContract]
```

```
int EndAdd(IAsyncResult res);
```

D. [OperationContract(AsyncPattern=true)]

```
IAsyncResult BeginAdd(int num1, int num2, AsyncCallback cb, object  
state);
```

```
int EndAdd(IAsyncResult res);
```

Answer: D

24. You create a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You create the following service contract.

```
<ServiceContract(> _
```

```
Public Interface IMath
```

```
    <OperationContract(> _
```

```
        Function Add(ByVal num1 As Integer, ByVal num2 As Integer) _
```

```
            As Integer
```

```
End Interface
```

You need to add an operation contract to perform the Add operation asynchronously.

Which operation contract should you use

A. <OperationContract(AsyncPattern:=True)> _

```
Function BeginAdd(ByVal num1 As Integer, ByVal num2 As Integer) _
```

```
As IAsyncResult
```

```
Function EndAdd(ByVal res As IAsyncResult) As Integer
```

B. <OperationContract(> _

```
Function BeginAdd(ByVal num1 As Integer, ByVal num2 As Integer, _
```

```
ByVal cb As AsyncCallback, ByVal state As Object) As Integer
```

```
Function EndAdd() As IAsyncResult
```

C. <OperationContract(> _

```
Function BeginAdd(ByVal num1 As Integer, ByVal num2 As Integer) _
```

```
As IAsyncResult
```

```
<OperationContract(> _
```

```
Function EndAdd(ByVal res As IAsyncResult) As Integer
```

D. <OperationContract(AsyncPattern:=True)> _
Function BeginAdd(ByVal num1 As Integer, ByVal num2 As Integer, _
ByVal cb As AsyncCallback, ByVal state As Object) As IAsyncResult
Function EndAdd(ByVal res As IAsyncResult) As Integer

Answer: D

25. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You need to ensure that data sent in a SOAP header is in the following XML format.

```
<Data>  
  <string>String 1</string>  
  <string>String 2</string>  
  <string>String 3</string>  
</Data>
```

Which code segment should you use

A. [DataContract]

```
public class MyMessage  
{  
  [MessageHeader]  
  public string[] Data;  
}
```

B. [DataContract]

```
public class MyMessage  
{  
  [MessageHeaderArray]  
  public string[] Data;  
}
```

C. [DataContract]

```
public class MyMessage  
{
```

```
[MessageProperty]
public string[] Data;
}
```

D. [MessageContract]

```
public class MyMessage
{
    [MessageBodyMember (Order=0)]
    public string[] Data;
}
```

Answer: A

26. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You need to ensure that data sent in a SOAP header is in the following XML format.

```
<Data>
  <string>String 1</string>
  <string>String 2</string>
  <string>String 3</string>
</Data>
```

Which code segment should you use

A. <MessageContract()> _

```
Public Class MyMessage
```

```
  <MessageHeader()> _
```

```
  Public Data() As String
```

```
End Class
```

B. <MessageContract()> _

```
Public Class MyMessage
```

```
  <MessageHeaderArray()> _
```

```
  Public Data() As String
```

```
End Class
```

C. <MessageContract()> _

```
Public Class MyMessage
```

```
    <MessageProperty()> _
```

```
    Public Data() As String
```

```
End Class
```

D. <MessageContract()> _

```
Public Class MyMessage
```

```
    <MessageBodyMember(Order:=0)> _
```

```
    Public Data() As String
```

```
End Class
```

Answer: A

27. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You need to programmatically add the following endpoint definition to the service.

```
http://localhost:8000/ExamService/service
```

Which code segment should you use

A. String baseAddress="http: //localhost:8000/ExamService";

```
BasicHttpBinding binding1=new BasicHttpBinding();
```

```
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
```

```
{
```

```
    host.AddServiceEndpoint(typeof(IExam),binding1,baseAddress);
```

```
}
```

B. String baseAddress="http: //localhost:8000/ExamService/service";

```
BasicHttpBinding binding1=new BasicHttpBinding();
```

```
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
```

```
{
```

```
    host.AddServiceEndpoint(typeof(IExam),binding1,baseAddress);
```

```
}
```

C. String baseAddress="http: //localhost:8000/ExamService";

```
WsHttpBinding binding1=new WsHttpBinding();
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
{
    host.AddServiceEndpoint(typeof(IEexam),binding1,baseAddress);
}
```

D. String baseAddress="net.tcp: //localhost:8000/ExamService/service";

```
NetTcpBinding binding1=new NetTcpBinding();
using(ServiceHost host=new ServiceHost(typeof(ExamService)))
{
    host.AddServiceEndpoint(typeof(IEexam),binding1,baseAddress);
}
```

Answer: B

28. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You need to programmatically add the following endpoint definition to the service.

```
http://localhost:8000/ExamService/service
```

Which code segment should you use

A. Dim baseAddress As String = "http: //localhost:8000/ExamService"

```
Dim binding1 As New BasicHttpBinding()
```

```
Using host As New ServiceHost(GetType(ExamService))
```

```
    host.AddServiceEndpoint(GetType(IEexam), binding1, baseAddress)
```

```
End Using
```

B. Dim baseAddress As String = _ "http: //localhost:8000/ExamService/service"

```
Dim binding1 As New BasicHttpBinding()
```

```
Using host As New ServiceHost(GetType(ExamService))
```

```
    host.AddServiceEndpoint(GetType(IEexam), binding1, baseAddress)
```

```
End Using
```

C. Dim baseAddress As String = "http: //localhost:8000/ExamService"

```
Dim binding1 As New WsHttpBinding()
```



```
Using host As New ServiceHost(GetType(ExamService))
    host.AddServiceEndpoint(GetType(IEExam), binding1, baseAddress)
End Using
D. Dim baseAddress As String = _
"http: //localhost:8000/ExamService/service"
Dim binding1 As New NetTcpBinding()
Using host As New ServiceHost(GetType(ExamService))
    host.AddServiceEndpoint(GetType(IEExam), binding1, baseAddress)
End Using
```

Answer: B

29. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

```
01 Uri baseAddress=
02  new Uri (3 http://localhost:8000/ExamService3 );
03 Uri mexAddress=new Uri("");
04 ServiceHost serviceHost=new ServiceHost(typeof(
05  ExamService), baseAddress);
06 ServiceMetadataBehavior smb=
07  serviceHost.Description.Behaviors.
08 Find<ServiceMetadataBehavior>();
09 if (smb != null)
10 {
11
12 }
13 else
14 {
15  smb=new ServiceMetadataBehavior();
16  serviceHost.Description.Behaviors.Add(smb);
```

17 }

18 smb.HttpGetUrl=mexAddress;

19

When you browse to the base address of the service, you receive the following message: "Metadata publishing for this service is currently disabled."

You debug the code and discover that the `ServiceMetadataBehavior` behavior was previously nonexistent.

You need to enable metadata publishing.

What should you do

- A. Delete lines 15 and 16.
- B. Modify the code segment at line 03 in the following manner.

Uri mexAddress=new Uri("/service");

- C. Insert the following code segment at line 11.

smb.HttpGetEnabled=true;

- D. Insert the following code segment at line 19.

smb.HttpGetEnabled=true;

Answer: D

30. You are creating a Windows Communication Foundation service by using Microsoft .NET Framework 3.5.

You write the following code segment. (Line numbers are included for reference only.)

01 Dim baseAddress As Uri = _

 New Uri("http://localhost:8000/ExamService")

02 Dim mexAddress As New Uri("")

03 Dim _serviceHost As New ServiceHost(_

 GetType(ExamService), baseAddress)

04 Dim smb As ServiceMetadataBehavior = _

 _serviceHost.Description.Behaviors.Find(_

 Of ServiceMetadataBehavior)()

05 If smb Is Nothing Then

06

07 Else

08 smb = New ServiceMetadataBehavior()

09 _serviceHost.Description.Behaviors.Add(smb)

10 End If

11 smb.HttpGetUrl = mexAddress

12

When you browse to the base address of the service, you receive the following message: "Metadata publishing for this service is currently disabled."

You debug the code and discover that the ServiceMetadataBehavior behavior was previously nonexistent.

You need to enable metadata publishing.

What should you do

- A. Delete lines 08 and 09.
- B. Modify the code segment at line 02 in the following manner.

```
Dim mexAddress As New Uri("/service")
```

- C. Insert the following code segment at line 06.

```
smb.HttpGetEnabled = True
```

- D. Insert the following code segment at line 12.

```
smb.HttpGetEnabled = True
```

Answer: D