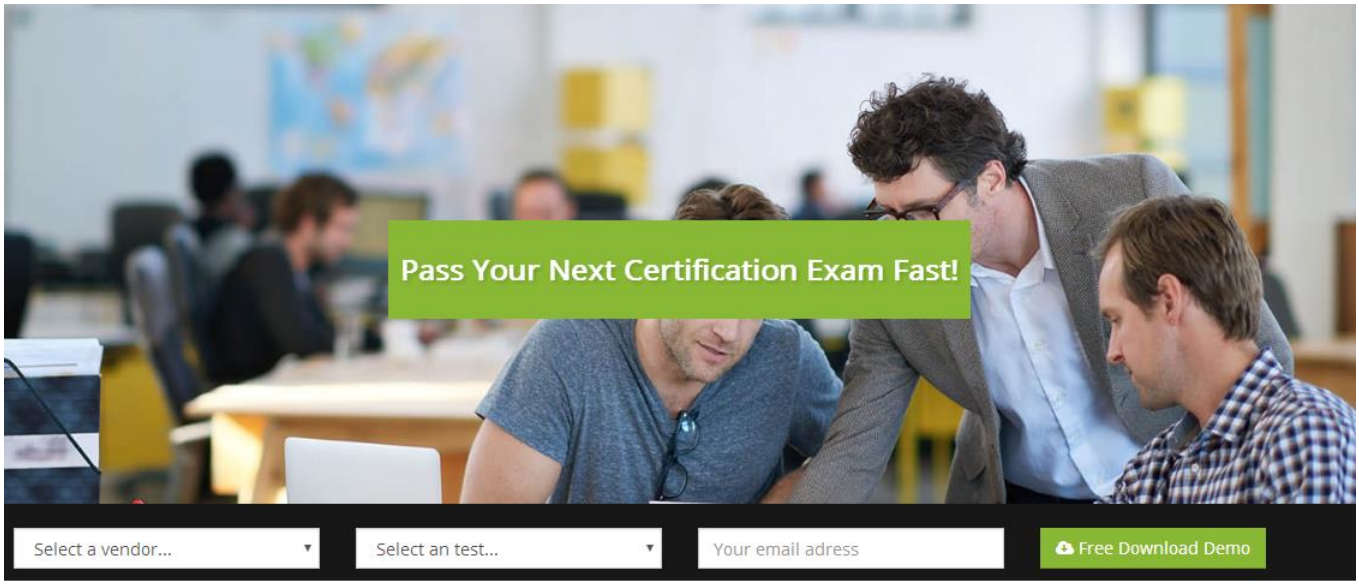
ITPassLeader



Instant Download | 365 Days Free Updates | Money Back Guarantee | Security & Privacy

Choose the version that fits your needs	PDF Version	Desktop Test Engine	Online Test Engine
Latest and Up-to-Date exam dumps with real exam questions answers.	✓	✓	✓
Get 12-Months free updates without any extra charges.	✓	✓	✓
Experience same exam environment before appearing in the certification exam.	✗	✓	✓
100% exam passing guarante in the first attempt.	✓	✓	✓
20% discount on more than one license and 30% discount on 5+ license purchases.	✗	✓	✓
100% secure purchase on SSL.	✓	✓	✓
Completely private purchase without sharing your personal info with anyone.	✓	✓	✓

http://www.itpassleader.com

High-praise Exam Dumps Questions grant you success by high pass rate - ITPassLeader

Exam : 070-513

Title : TS: Windows Communication Foundation velopment with Microsoft .NET Framework 4

Vendors : Microsoft

Version : DEMO

NO.1 new NetTcpBinding(),

NO.2 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

NO.3 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

NO.4 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">
<enableWebScript />
</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

NO.5 new EndpointAddress("net.tcp://localhost:8080/Logger")

NO.6 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 lsRequired property of each data member to true.
Answer: D

NO.7 RoutingConfiguration rc = new RoutingConfiguration();

NO.8 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

NO.9 You are creating a Windows Communication Foundation (WCF) service. You have the following
requirements:
Messages must be sent over TCP
The service must support transactions.
Messages must be encoded using a binary encoding
Messages must be secured using Windows stream-based security.
You need to implement a custom binding for the service. In which order should the binding stack be
configured?
A.tcpTransport, windowsStreamSecurity, transactionFlow, binaryMessageEncoding
B.transactionFlow, binaryMessageEncoding, windowsStreamSecurity, tcpTransport
C.windowsStreamSecurity, tcpTransport, binaryMessageEncoding, transactionFlow
D.binaryMessageEncoding, transactionFlow, tcpTransport, windowsStreamSecurity

Answer: B

NO.10 A Windows Communication Foundation (WCF) service implements a contract with one-way and
request-reply operations.
The service is exposed over a TCP transport. Clients use a router to communicate with the service.
The router is implemented as follows. (Line numbers are included for reference only.)

```
01 ServiceHost host = new ServiceHost(typeof(RoutingService));
02 host.AddServiceEndpoint(
03     typeof(ISimplexDatagramRouter),
04     new NetTcpBinding(), "net.tcp://localhost/Router"
05    );
06 List<ServiceEndpoint> lep = new List<ServiceEndpoint>();
07 lep.Add(
08     new ServiceEndpoint(
09         ContractDescription.GetContract(
10             typeof(ISimplexDatagramRouter)
11    ),
```

NO.11 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

NO.12 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

NO.13 rc.FilterTable.Add(new MatchAllMessageFilter(), lep);

NO.14)

NO.15 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

NO.16 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="Arithmetic")]
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

NO.17 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

NO.18 host.Description.Behaviors.Add(new RoutingBehavior(rc));
Request-reply operations are failing. You need to ensure that the router can handle one-way and
request-reply operations.
What should you do?
A.Change line 03 as follows:
typeof(IRequestReplyRouter),
B.Change line 03 as follows:
typeof(IDuplexSessionRouter),
C.Change line 10 as follows:
typeof(IRequestReplyRouter)
D.Change line 10 as follows:
typeof(IDuplexSessionRouter)
Answer: B
12.You are modifying an existing Windows Communication Foundation (WCF) service that is defined as
follows:
[ServiceContract]
public interface IMessageProcessor
{

[OperationContract]
void ProcessMessages();
}
public class MessageProcessor: IMessageProcessor
{
public void ProcessMessage();
SubmitOrder();
}
SubmitOrder makes a call to another service. The ProcessMessage method does not perform as
expected under a heavy load.
You need to enable processing of multiple messages. New messages must only be processed when the
ProcessMessage method is not processing requests,
or when it is waiting for calls to SubmitOrder to return.
Which attribute should you apply to the MessageProcessor class?
A.CallbackBehavior(ConcurrencyMode=ConcurencyMode.Reentrant)
B.CallbackBehavior(ConcurrencyMode=ConcurrencyMode.Multiple)
C.ServiceBehavior(ConcurrencyMode=ConcurrencyMode.Reentrant)
D.ServiceBehavior(ConcurrencyMode=ConcurrencyMode.Multiple)
Answer: C
13.A Windows Communication Foundation (WCF) service listens for messages at net.tcp://www.contoso.com/MyService.
It has a logical address at http://www.contoso.com/MyService. The configuration for the WCF client is as
follows:
<endpoint address="http://www.contoso.com/MyService"
binding="netTcpBinding"
bindingConfiguraton="NetTcpBinding_IMyService"
contract="ServiceReference1.IMyService"
name="NetTcpBinding_IMyService"/>
The generated configuration does not provide enough information for the client to communicate with the
server.
You need to update the client so that it can communicate with the server. What should you do?
A.In the client configuration, change the value of the address attribute to net.tcp://www.contoso.com/MyService

B.In the client configuration, change the value of the address attribute to net.tcp://www.contoso.com/MyService listen=http://www.contoso.com/MyService.
C.After instantiating the client and before invoking any service operation, add this line of code.
EndpointBehaviors.Add(new EndpointDiscoveryBehavior(){ Enabled = true });
D.After instantiating the client and before invoking any service operation, add this line of code.
client.Endpoint.Behaviors.Add(new ClientViaBehavior(new Uri("net.tcp://www.contoso.com/IMyService")));
Answer: D
14.A Windows Communication Foundation (WCF) service is self-hosted in a console application.
The service implements the IDataAccess contract, which is defined in the MyApplication namespace.
The service is implemented in a class named DataAccessService which implements the IDataAccess
interface and also is defined in the MyApplication namespace. The hosting code is as follows.
(Line numbers are included for reference only.)

```
01 static void Main(string[] args)
02 {
03     ServiceHost host;
04     ...
05     host.Open();
06     Console.ReadLine();
07     host.Close();
08 }
```

You need to create a ServiceHost instance and assign it to the host variable. You also need to instantiate
the service host.
Which line of code should you insert at line 04?
A.host = new ServiceHost("MyApplication.DataAccessService");
B.host = new ServiceHost("MyApplication.DataAccess");
C.host = new ServiceHost(typeof(IDataAccess));
D.host = new ServiceHost(typeof(DataAccessService));
Answer: D
15.A Windows Communication Foundation (WCF) service implements the following contract.

```
[ServiceContract]
```

```
public interface IHelloService
{
[OperationContract(WebGet(UriTemplate="hello?name={name}"))]
string SayHello(string name);
}
```

The implementation is as follows:

```
public class HelloService: IHelloService
{
public string SayHello(string name)
{
return "Hello " + name;
}
}
```

The service is self-hosted, and the hosting code is as follows:

```
WebServiceHost svcHost = CreateHost();
svcHost.Open();
Console.ReadLine();
svcHost.Close();
```

You need to implement CreateHost so that the service has a single endpoint hosted at http://localhost:8000/HelloService.

Which code segment should you use?

A.

```
WebServiceHost svcHost = new WebServiceHost(typeof(HelloService));
svcHost.AddServiceEndpoint(typeof(IHelloService),
new WebHttpBinding(WebHttpSecurityMode.None),
"http://localhost:8000/HelloService");
return svcHost;
```

B.

```
Uri baseAddress = new Uri("http://localhost:8000");
WebServiceHost svcHost = new WebServiceHost(typeof(HelloService), baseAddress);
svcHost.AddServiceEndpoint(typeof(IHelloService),
new WebHttpBinding(WebHttpSecurityMode.None),
"HelloService");
return svcHost;
```

C.

```
WebServiceHost svcHost = new WebServiceHost(new HelloService());
svcHost.AddServiceEndpoint(typeof(IHelloService),
new WebHttpBinding(WebHttpSecurityMode.None),
"http://localhost:8000/HelloService");
return svcHost
```

D.

```
Uri baseAddress = new Uri("http://localhost:8000/");
```

```
WebServiceHost svcHost = new WebServiceHost(new HelloService(), baseAddress);
svcHost.AddServiceEndpoint(typeof(IHelloService),
new WebHttpBinding(WebHttpSecurityMode.None),
"HelloService");
return svcHost;
```

Answer: B

16.You are building a client for a Windows Communication Foundation (WCF) service.
You need to create a proxy to consume this service. Which class should you use?
A.ChannelFactory<TChannel>
B.ServiceHost
C.ClientRuntime
D.CommunicationObject
Answer: A

17.You are working with a Windows Communication Foundation (WCF) client application that has a
generated proxy named SampleServiceProxy.
When the client application is executing, in line 04 of the following code, the channel faults (Line numbers
are included for reference only.)

```
01 SampleServiceProxy proxy = new SampleServiceProxy();
02 try
03 {
04     proxy.ProcessInvoice(invoice);
05 }
06 catch
07 {
08     if(proxy.State == CommunicationState.Faulted)
09     {
10         ...
11     }
12 }
13 proxy.UpdateCustomer(customer);
```

You need to return proxy to a state in which it can successfully execute the call in line 13.
Which code segment should you use at line 10?
A.proxy.Close();
B.proxy = new SampleServiceProxy();
C.proxy.Abort();
D.proxy.Open();

Answer: B
18.A Windows Communication Foundation (WCF) service has a callback contract. You are developing a
client application that will call this service.
You must ensure that the client application can interact with the WCF service. What should you do?
A.On the OperationContractAttribute, set the AsyncPattern property value to true.
B.On the OperationContractAttribute, set the ReplyAction property value to the endpoint address of the
client.
C.On the client, create a proxy derived from DuplexClientBase<TChannel>.
D.On the client, use GetCallbackChannel<T>.
Answer: C

NO.19);

NO.20 A Windows Communication Foundation (WCF) client configuration file contains the following XML
segment in the system.serviceModel element.
<client>
<endpoint address="net.tcp://server/ContosoService"
binding="netTcpBinding"
contract="Contoso.IContosoService"
name="netTcp"/>
<endpoint address="net.pipe://localhost/ContosoService"
binding="netNamedPipeBinding"
contract="Contoso.IContosoService"
name="netPipe" />
</client>
You need to create a channel factory that can send messages to the endpoint listening at net.pipe://localhost/ContosoService.
Which code segment should you use?
A.ChannelFactory<Contoso.IContoso> factory = new
ChannelFactory<Contoso.IContoso>("Contoso.IContoso");
B.ChannelFactory<Contoso.IContoso> factory = new
ChannelFactory<Contoso.IContoso>("netNamedPipeBinding");
C.ChannelFactory<Contoso.IContoso> factory = new
ChannelFactory<Contoso.IContoso>("netPipe");

D.ChannelFactory<Contoso.IContoso> factory = new
ChannelFactory<Contoso.IContoso>("net.pipe//localhost/ContosoService");
Answer: C