

WSD Lab-4

Aim : Creating basic WCF service and Implementing a client application to consume the service

1) Create demo WCF service (WCF Service Library) and test it using WCF test client.

SOAP messages

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.
org/IService1/GetData</Action>
  </s:Header>
  <s:Body>
    <GetData xmlns="http://tempuri.org/">
      <value>9</value>
    </GetData>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetDataResponse xmlns="http://tempuri.org/">
      <GetDataResult>You entered: 9</GetDataResult>
    </GetDataResponse>
  </s:Body>
</s:Envelope>
```

WSDL contents

```

<wsdl:definitions name="Service1" targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://tempuri.org/Imports">
      <xsd:import
        schemaLocation="http://localhost:8733/Design_Time_Addresses/WCFServiceDemo/Service1/?xsd=xsd0" namespace="http://tempuri.org"/>
      <xsd:import
        schemaLocation="http://localhost:8733/Design_Time_Addresses/WCFServiceDemo/Service1/?xsd=xsd1"
        namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xsd:import
        schemaLocation="http://localhost:8733/Design_Time_Addresses/WCFServiceDemo/Service1/?xsd=xsd2"
        namespace="http://schemas.datacontract.org/2004/07/WCFServiceDemo" />
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="IService1_GetData_InputMessage">
    <wsdl:part name="parameters" element="tns:GetData"/>
  </wsdl:message>
  <wsdl:message name="IService1_GetData_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetDataResponse"/>
  </wsdl:message>
  <wsdl:message name="IService1_GetDataUsingDataContract_InputMessage">
    <wsdl:part name="parameters" element="tns:GetDataUsingDataContract"/>
  </wsdl:message>
  <wsdl:message name="IService1_GetDataUsingDataContract_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetDataUsingDataContractResponse"/>
  </wsdl:message>
  <wsdl:portType name="IService1">
    <wsdl:operation name="GetData">
      <wsdl:input wsaw:Action="http://tempuri.org/IService1/GetData"
        message="tns:IService1_GetData_InputMessage"/>
      <wsdl:output wsaw:Action="http://tempuri.org/IService1/GetDataResponse"
        message="tns:IService1_GetData_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="GetDataUsingDataContract">

```

```

<wsdl:input wsaw:Action="http://tempuri.org/IService1/GetDataUsingDataContract"
message="tns:IService1_GetDataUsingDataContract_InputMessage"/>
<wsdl:output
wsaw:Action="http://tempuri.org/IService1/GetDataUsingDataContractResponse"
message="tns:IService1_GetDataUsingDataContract_OutputMessage"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BasicHttpBinding_IService1" type="tns:IService1">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="GetData">
<soap:operation soapAction="http://tempuri.org/IService1/GetData"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetDataUsingDataContract">
<soap:operation soapAction="http://tempuri.org/IService1/GetDataUsingDataContract"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Service1">
<wsdl:port name="BasicHttpBinding_IService1"
binding="tns:BasicHttpBinding_IService1">
<soap:address
location="http://localhost:8733/Design_Time_Addresses/WCFServiceDemo/Service1/"
>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

service and client code**Service:**

Iservice.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace WCFServiceDemo
{
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        string GetData(int value);

        [OperationContract]
        CompositeType GetDataUsingDataContract(CompositeType composite);
    }

    [DataContract]
    public class CompositeType
    {
        bool boolValue = true;
        string stringValue = "Hello ";

        [DataMember]
        public bool BoolValue
        {
            get { return boolValue; }
            set { boolValue = value; }
        }
    }
}
```

```

[DataMember]
public string StringValue
{
    get { return stringValue; }
    set { stringValue = value; }
}
}
}

```

Service1.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace WCFServiceDemo
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the
    class name "Service1" in both code and config file together.
    public class Service1 : IService1
    {
        public string GetData(int value)
        {
            return string.Format("You entered: {0}", value);
        }

        public CompositeType GetDataUsingDataContract(CompositeType composite)
        {
            if (composite == null)
            {
                throw new ArgumentNullException("composite");
            }
            if (composite.BoolValue)
            {

```

```
        composite.StringValue += "Suffix";
    }
    return composite;
}
}
```

Client:

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ClientApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void button1_Click(object sender, EventArgs e)
            {
                ServiceReference1.Service1Client sc = new ServiceReference1.Service1Client();
                label1.Text = sc.GetData(int.Parse(textBox1.Text));
            }
        }
    }
}
```

config files

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<configuration>
```

```
<appSettings>
```

```
<add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
```

```
</appSettings>
```

```
<system.web>
```

```
<compilation debug="true" />
```

```
</system.web>
```

<!-- When deploying the service library project, the content of the config file must be added to the host's

app.config file. System.Configuration does not support config files for libraries. -->

```
<system.serviceModel>
```

```
<services>
```

```
<service name="WCFServiceDemo.Service1">
```

```
<host>
```

```
<baseAddresses>
```

```
<add baseAddress =
```

```
"http://localhost:8733/Design_Time_Addresses/WCFServiceDemo/Service1/" />
```

```
</baseAddresses>
```

```
</host>
```

```
<!-- Service Endpoints -->
```

```
<!-- Unless fully qualified, address is relative to base address supplied above -->
```

```
<endpoint address="" binding="basicHttpBinding"
```

```
contract="WCFServiceDemo.IService1">
```

```
<!--
```

Upon deployment, the following identity element should be removed or replaced to reflect the

identity under which the deployed service runs. If removed, WCF will infer an appropriate identity

automatically.

```
-->
```

```
<identity>
```

```
<dns value="localhost"/>
```

```
</identity>
```

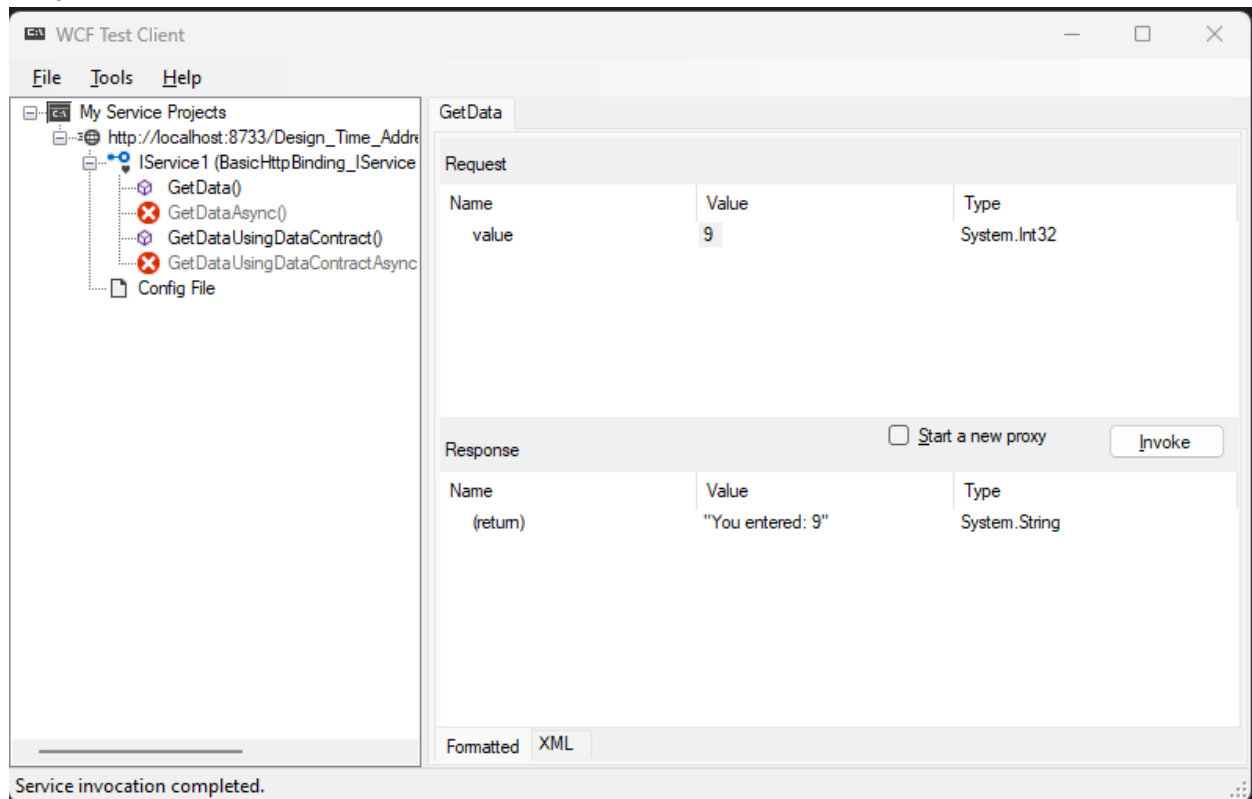
```
</endpoint>
```

```
<!-- Metadata Endpoints -->
```

```
<!-- The Metadata Exchange endpoint is used by the service to describe itself to
clients. -->
<!-- This endpoint does not use a secure binding and should be secured or removed
before deployment -->
<endpoint address="mex" binding="mexHttpBinding"
contract="IMetadataExchange"/>
</service>
</services>
<behaviors>
<serviceBehaviors>
<behavior>
<!-- To avoid disclosing metadata information,
set the values below to false before deployment -->
<serviceMetadata httpGetEnabled="True" httpsGetEnabled="True"/>
<!-- To receive exception details in faults for debugging purposes,
set the value below to true. Set to false before deployment
to avoid disclosing exception information -->
<serviceDebug includeExceptionDetailInFaults="False" />
</behavior>
</serviceBehaviors>
</behaviors>
</system.serviceModel>

</configuration>
```

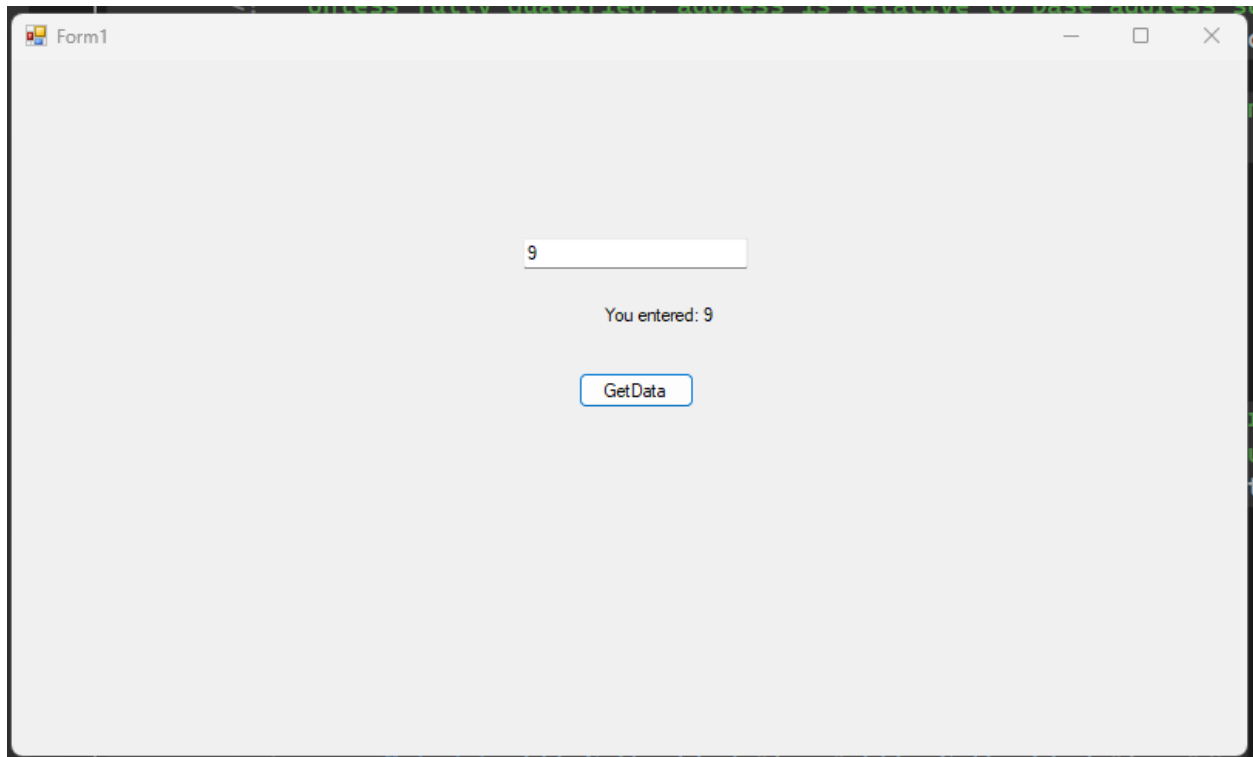

output screenshots



3) Develop a windows form(.NET Framework) based client application to consume the demo service.

Config File:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="BasicHttpBinding IService1" />
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint
address="http://localhost:8733/Design_Time_Addresses/WCFServiceDemo/Service1/"
        binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding IService1"
        contract="ServiceReference1.IService1" name="BasicHttpBinding IService1" />
    </client>
  </system.serviceModel>
</configuration>
```



The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main area of the form is light gray. In the center, there is a text input field containing the number "9". Below the input field, the text "You entered: 9" is displayed. At the bottom center, there is a button labeled "GetData".

- 4) Create a calculator service that supports four basic arithmetic operations (add, subtract, multiply, divide). Test the service using WCF test client.
- 5) Develop a windows form(.NET Framework) based client application to consume the calculator service.

SOAP messages

Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://tempuri.
org/IService1/Addition</Action>
  </s:Header>
  <s:Body>
    <Addition xmlns="http://tempuri.org/">
      <a>7</a>
      <b>9</b>
    </Addition>
  </s:Body>
</s:Envelope>
```

Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <AdditionResponse xmlns="http://tempuri.org/">
      <AdditionResult>16</AdditionResult>
    </AdditionResponse>
  </s:Body>
</s:Envelope>
```

WSDL contents

```

<wsdl:definitions name="Service1" targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <xsd:schema targetNamespace="http://tempuri.org/Imports">
      <xsd:import
        schemaLocation="http://localhost:8733/Design_Time_Addresses/Calculator/Service1/
        ?xsd=xsd0" namespace="http://tempuri.org/" />
      <xsd:import
        schemaLocation="http://localhost:8733/Design_Time_Addresses/Calculator/Service1/
        ?xsd=xsd1" namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="IService1_GetData_InputMessage">
    <wsdl:part name="parameters" element="tns:GetData" />
  </wsdl:message>
  <wsdl:message name="IService1_GetData_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetDataResponse" />
  </wsdl:message>
  <wsdl:message name="IService1_Addition_InputMessage">
    <wsdl:part name="parameters" element="tns:Addition" />
  </wsdl:message>
  <wsdl:message name="IService1_Addition_OutputMessage">
    <wsdl:part name="parameters" element="tns:AdditionResponse" />
  </wsdl:message>
  <wsdl:message name="IService1_Subtraction_InputMessage">
    <wsdl:part name="parameters" element="tns:Subtraction" />
  </wsdl:message>
  <wsdl:message name="IService1_Subtraction_OutputMessage">
    <wsdl:part name="parameters" element="tns:SubtractionResponse" />
  </wsdl:message>
  <wsdl:message name="IService1_Multiplication_InputMessage">
    <wsdl:part name="parameters" element="tns:Multiplication" />
  </wsdl:message>
  <wsdl:message name="IService1_Multiplication_OutputMessage">
    <wsdl:part name="parameters" element="tns:MultiplicationResponse" />
  </wsdl:message>
  <wsdl:message name="IService1_Division_InputMessage">
    <wsdl:part name="parameters" element="tns:Division" />
  </wsdl:message>

```

```

</wsdl:message>
<wsdl:message name="IService1_Division_OutputMessage">
<wsdl:part name="parameters" element="tns:DivisionResponse"/>
</wsdl:message>
<wsdl:portType name="IService1">
<wsdl:operation name="GetData">
<wsdl:input wsaw:Action="http://tempuri.org/IService1/GetData"
message="tns:IService1_GetData_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/IService1/GetDataResponse"
message="tns:IService1_GetData_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="Addition">
<wsdl:input wsaw:Action="http://tempuri.org/IService1/Addition"
message="tns:IService1_Addition_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/IService1/AdditionResponse"
message="tns:IService1_Addition_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="Subtraction">
<wsdl:input wsaw:Action="http://tempuri.org/IService1/Subtraction"
message="tns:IService1_Subtraction_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/IService1/SubtractionResponse"
message="tns:IService1_Subtraction_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="Multiplication">
<wsdl:input wsaw:Action="http://tempuri.org/IService1/Multiplication"
message="tns:IService1_Multiplication_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/IService1/MultiplicationResponse"
message="tns:IService1_Multiplication_OutputMessage"/>
</wsdl:operation>
<wsdl:operation name="Division">
<wsdl:input wsaw:Action="http://tempuri.org/IService1/Division"
message="tns:IService1_Division_InputMessage"/>
<wsdl:output wsaw:Action="http://tempuri.org/IService1/DivisionResponse"
message="tns:IService1_Division_OutputMessage"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BasicHttpBinding_IService1" type="tns:IService1">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="GetData">

```

```
<soap:operation soapAction="http://tempuri.org/IService1/GetData"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Addition">
<soap:operation soapAction="http://tempuri.org/IService1/Addition"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Subtraction">
<soap:operation soapAction="http://tempuri.org/IService1/Subtraction"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Multiplication">
<soap:operation soapAction="http://tempuri.org/IService1/Multiplication"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
```

```
<wsdl:operation name="Division">
<soap:operation soapAction="http://tempuri.org/IService1/Division"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Service1">
<wsdl:port name="BasicHttpBinding_IService1"
binding="tns:BasicHttpBinding_IService1">
<soap:address
location="http://localhost:8733/Design_Time_Addresses/Calculator/Service1/">
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```


service and client code**IService1.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;
```

```
namespace Calculator
```

```
{
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        string GetData(int value);

        [OperationContract]
        int Addition(int a, int b);

        [OperationContract]
        int Subtraction(int a, int b);

        [OperationContract]
        int Multiplication(int a, int b);

        [OperationContract]
        decimal Division(int a, int b);
    }
}
```

Service1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

namespace Calculator
{
    public class Service1 : IService1
    {
        public string GetData(int value)
        {
            return string.Format("You entered: {0}", value);
        }

        public int Addition(int a , int b) {
            return a + b;
        }
        public int Subtraction(int a, int b)
        {
            return a - b;
        }
        public int Multiplication(int a, int b)
        {
            return a * b;
        }
        public decimal Division(int a, int b)
        {
            return (decimal)a / b;
        }
    }
}
```

Client:

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CalcClient
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void CalculateBtn_Click(object sender, EventArgs e)
        {
            int a = int.Parse(textBox1.Text);
            int b = int.Parse(textBox2.Text);
            Answerlbl.Text = "Answer = ";
            ServiceReference1.Service1Client sc = new ServiceReference1.Service1Client();
            if (add.Checked)
            {
                Answerlbl.Text += sc.Addition(a, b).ToString();
            }
            if (sub.Checked)
            {
                Answerlbl.Text += sc.Subtraction(a, b).ToString();
            }
        }
    }
}
```

```

    }
    if (mul.Checked)
    {
        Answerlbl.Text += sc.Multiplication(a, b).ToString();
    }
    if (div.Checked)
    {
        Answerlbl.Text += sc.Division(a, b).ToString();
    }
}
}
}

```

config files

Service configurationfile :

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" />
  </system.web>
  <!-- When deploying the service library project, the content of the config file must be
added to the host's
  app.config file. System.Configuration does not support config files for libraries. -->
  <system.serviceModel>
    <services>
      <service name="Calculator.Service1">
        <host>
          <baseAddresses>
            <add baseAddress =
"http://localhost:8733/Design_Time_Addresses/Calculator/Service1/" />
          </baseAddresses>
        </host>
        <!-- Service Endpoints -->

```

```
<!-- Unless fully qualified, address is relative to base address supplied above -->
<endpoint address="" binding="basicHttpBinding" contract="Calculator.IService1">
<!--
```

Upon deployment, the following identity element should be removed or replaced to reflect the

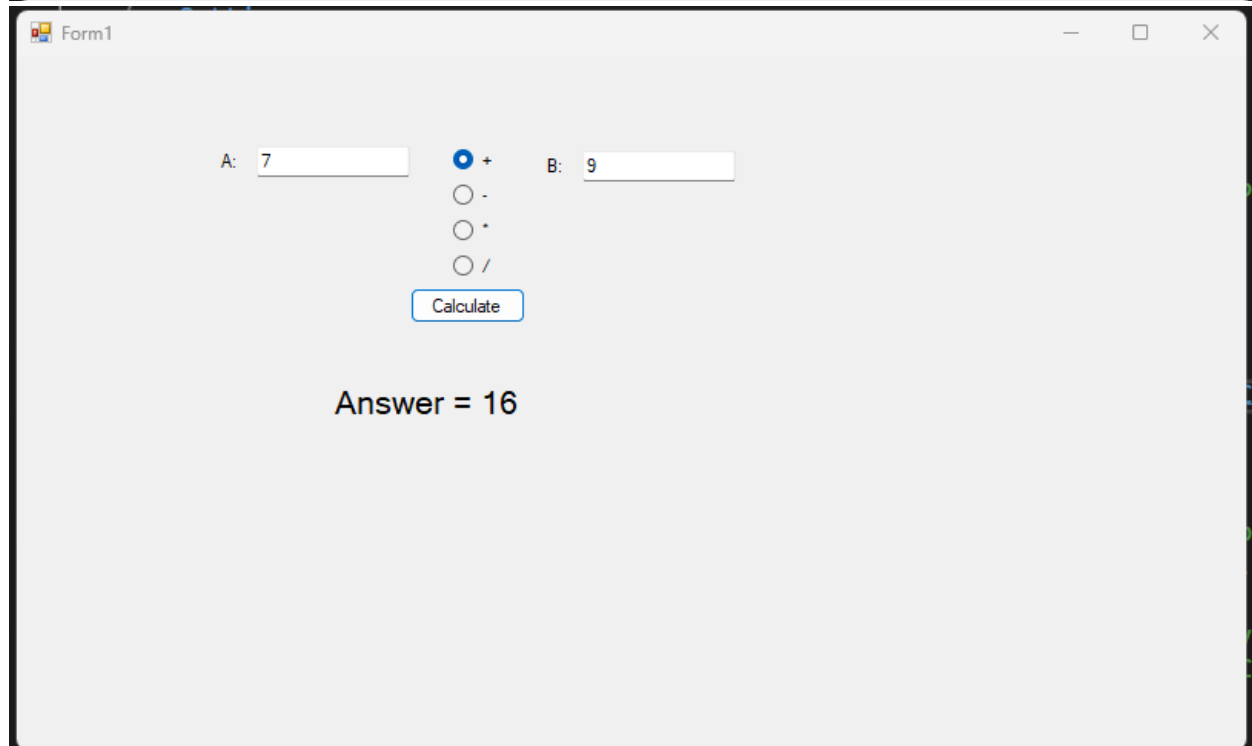
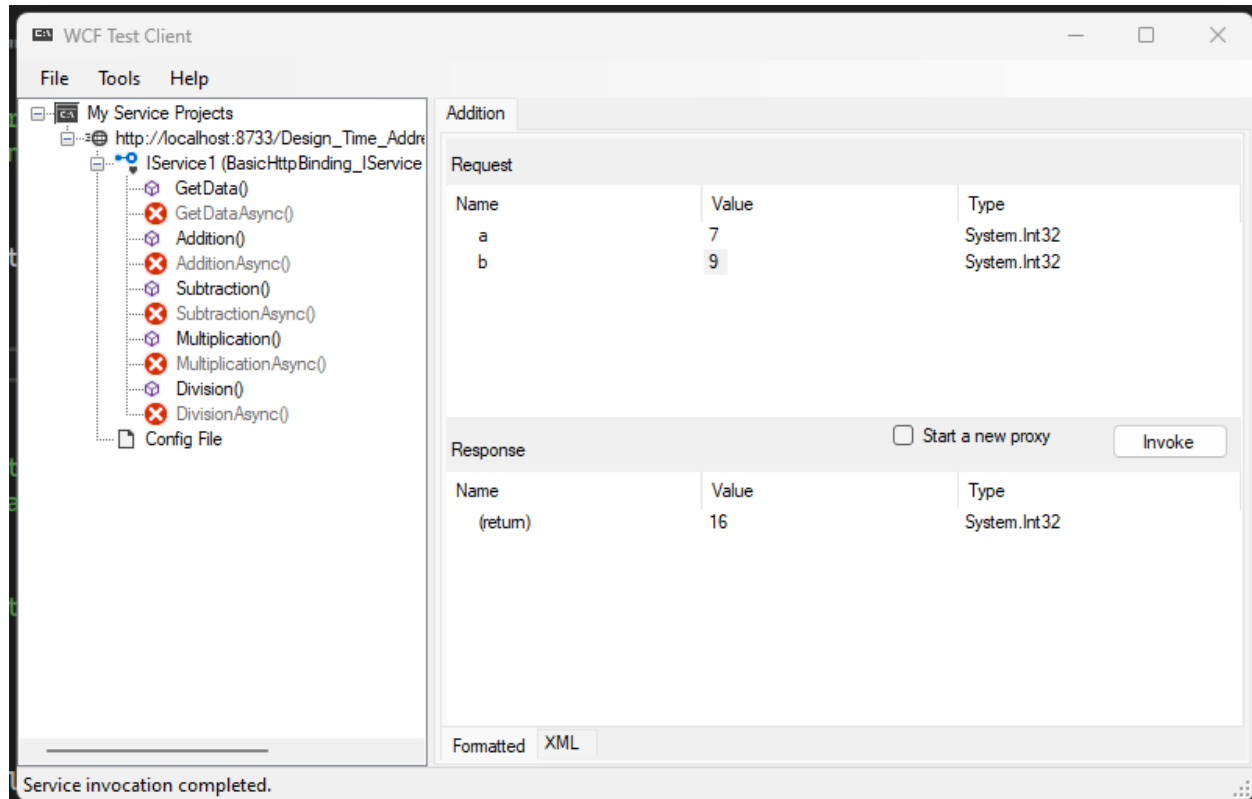
identity under which the deployed service runs. If removed, WCF will infer an appropriate identity automatically.

```
-->
<identity>
  <dns value="localhost"/>
</identity>
</endpoint>
<!-- Metadata Endpoints -->
<!-- The Metadata Exchange endpoint is used by the service to describe itself to
clients. -->
<!-- This endpoint does not use a secure binding and should be secured or removed
before deployment -->
<endpoint address="mex" binding="mexHttpBinding"
contract="IMetadataExchange"/>
</service>
</services>
<behaviors>
  <serviceBehaviors>
    <behavior>
      <!-- To avoid disclosing metadata information,
      set the values below to false before deployment -->
      <serviceMetadata httpGetEnabled="True" httpsGetEnabled="True"/>
      <!-- To receive exception details in faults for debugging purposes,
      set the value below to true. Set to false before deployment
      to avoid disclosing exception information -->
      <serviceDebug includeExceptionDetailInFaults="False" />
    </behavior>
  </serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>
```

Client Configuration file:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="BasicHttpBinding IService1" />
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint
address="http://localhost:8733/Design_Time_Addresses/Calculator/Service1/"
        binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding IService1"
        contract="ServiceReference1.IService1" name="BasicHttpBinding IService1" />
    </client>
  </system.serviceModel>
</configuration>
```

output screenshots



Form1

A: ☐ + B:
☒ -
☐ *
☐ /

Answer = -2

Form1

A: ☐ + B:
☐ -
☒ *
☐ /

Answer = 63

Form1

A:

☐ +
☐ -
☐ *
☒ /

B:

Answer = 0.777777777777778

6) Develop the web client to consume the calculator service.

.aspx.cs file

```
using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebClient
{

    public partial class WebForm1 : System.Web.UI.Page
    {

        ServiceReference1.IcalculatorClient service;

        protected void Page_Load(object sender, EventArgs e)
        {

            service = new ServiceReference1.IcalculatorClient();

        }

        protected void Button1_Click(object sender, EventArgs e)
        {

            int s1 = int.Parse(TextBox1.Text);

            int s2 = int.Parse(TextBox2.Text);

            double ans = new double();
```

```
switch (DropDownList1.SelectedItem.ToString()) {  
    case "+":  
        ans = service.add(s1, s2);  
        break;  
    case "-":  
        ans = service.subtract(s1, s2);  
        break;  
    case "*":  
        ans = service.multiply(s1, s2);  
        break;  
    case "/":  
        ans = service.divide(s1, s2);  
        break;  
    }  
    Label1.Text = ans.ToString();  
}  
}
```

.aspx file

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebClient.WebForm1" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title></title>
```

```
    <style type="text/css">
```

```
        .auto-style1 {
```

```
            width: 100%;
```

```
            height: 172px;
```

```
        }
```

```
    </style>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div>
```

```
            <table class="auto-style1">
```

```
                <tr>
```

```
                    <td>
```

```
                        <asp:TextBox ID="TextBox1" runat="server" Font-
Size="Medium"></asp:TextBox>
```

```
                    </td>
```

```
                <td>
```

```

Large">
    <asp:DropDownList ID="DropDownList1" runat="server" Font-Size="XX-
    <asp:ListItem>+</asp:ListItem>
    <asp:ListItem>-</asp:ListItem>
    <asp:ListItem>*</asp:ListItem>
    <asp:ListItem>/</asp:ListItem>
    </asp:DropDownList>
</td>
<td>
    <asp:TextBox ID="TextBox2" runat="server" Font-
Size="Medium"></asp:TextBox>
</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    </td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>

```

```
<asp:Button ID="Button1" runat="server" Font-Bold="True" Font-  
Size="Large" OnClick="Button1_Click" Text="Calculate" />
```

```
</td>
```

```
<td>&nbsp;</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

Web.config file:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!--
```

For more information on how to configure your ASP.NET application, please visit

<https://go.microsoft.com/fwlink/?LinkId=169433>

```
-->
```

```
<configuration>
```

```
<system.web>
```

```
<compilation debug="true" targetFramework="4.7.2" />
```

```
<httpRuntime targetFramework="4.7.2" />
```

```
</system.web>
```

```
<system.codedom>
```

```
<compilers>
```

```
<compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
```

```
<compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008
/define:_MYTYPE=\"Web\" /optionInfer+" />
```

```
</compilers>
```

```
</system.codedom>

<system.serviceModel>

  <bindings>

    <basicHttpBinding>

      <binding name="BasicHttpBinding_Icalculator" />

    </basicHttpBinding>

  </bindings>

  <client>

    <endpoint
address="http://localhost:8733/Design_Time_Addresses/Calculator/Calculator/"

      binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding_Icalculator"
      contract="ServiceReference1.Icalculator" name="BasicHttpBinding_Icalculator"
/>

  </client>

</system.serviceModel>

</configuration>
```


Output:

<input type="text" value="13"/>	<div>/ ▾</div>	<input type="text" value="3"/>
4.333333333333333		
<div>Calculate</div>		

<input type="text" value="13"/>	<div>* ▾</div>	<input type="text" value="3"/>
39		
<div>Calculate</div>		