Name : Karangiya Mihirkumar Parbatbhai

Lab : 05

Roll No : CE020

Batch : A1

1. **Calculator (Addition , Subtraction) of complex numbers (real part, imaginary part) using data contract.**

   Iservice1.cs :

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

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

        [OperationContract]
        ComplexNumber Addition(ComplexNumber a,
ComplexNumber b);
        [OperationContract]
        ComplexNumber Subtraction(ComplexNumber a,
ComplexNumber b);

        }
```

```
    [DataContract]
    public class ComplexNumber
    {
    [DataMember]
    public double real { get; set; }
    [DataMember]
    public double imaginary { get; set; }

    }
}
```

Service1.cs :

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

namespace WcfDataContractCalc
{

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

```
            public ComplexNumber Addition(ComplexNumber a,
ComplexNumber b)
            {
            ComplexNumber temp = new ComplexNumber();
            temp.real = a.real + b.real;
            temp.imaginary = a.imaginary + b.imaginary;

            return temp;
            }

            public ComplexNumber Subtraction(ComplexNumber a,
ComplexNumber b)
            {
            ComplexNumber temp = new ComplexNumber();
            temp.real = a.real - b.real;
            temp.imaginary = a.imaginary - b.imaginary;

            return temp;
            }

            }
      }
```

Config file of service :

```
<?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="WcfDataContractCalc.Service1">
            <host>
            <baseAddresses>
            <add baseAddress =
"http://localhost:8733/Design_Time_Addresses/WcfDataContract
Calc/Service1/" />
            </baseAddresses>
            </host>
            <!-- Service Endpoints -->
            <!-- Unless fully qualified, address is relative to base
address supplied above -->
            <endpoint address="" binding="basicHttpBinding"
contract="WcfDataContractCalc.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 code :

```
using ComplexClient.ServiceReference1;
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 ComplexClient
{
    public partial class Form1 : Form
    {
    public Form1()
    {
    InitializeComponent();
    }
```

```
            private void Add_Click(object sender, EventArgs e)
            {
            ServiceReference1.Service1Client sc = new
ServiceReference1.Service1Client();
            ServiceReference1.ComplexNumber a = new
ServiceReference1.ComplexNumber();
            a.real = Double.Parse(realBox1.Text);
            a.imaginary = Double.Parse(imagBox1.Text);

            ServiceReference1.ComplexNumber b = new
ServiceReference1.ComplexNumber();
            b.real = Double.Parse(realBox2.Text);
            b.imaginary = Double.Parse(imagBox2.Text);

            ServiceReference1.ComplexNumber c = new
ServiceReference1.ComplexNumber();

            c = sc.Addition(a,b);

            realans.Text = c.real.ToString();
            imagans.Text = c.imaginary.ToString();
            }

            private void Sub_Click(object sender, EventArgs e)
            {
            ServiceReference1.Service1Client sc = new
ServiceReference1.Service1Client();
            ServiceReference1.ComplexNumber a = new
ServiceReference1.ComplexNumber();
```

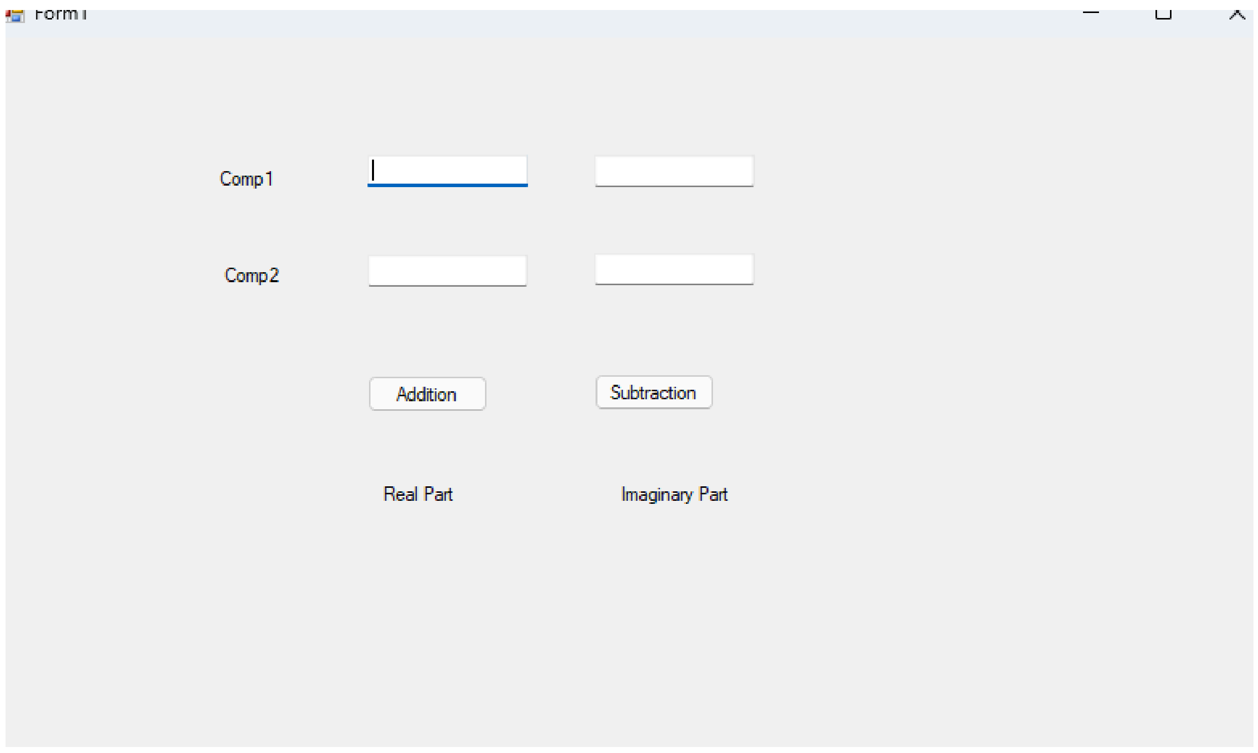
```
            a.real = Double.Parse(realBox1.Text);
            a.imaginary = Double.Parse(imagBox1.Text);

            ServiceReference1.ComplexNumber b = new
ServiceReference1.ComplexNumber();
            b.real = Double.Parse(realBox2.Text);
            b.imaginary = Double.Parse(imagBox2.Text);

            ServiceReference1.ComplexNumber c = new
ServiceReference1.ComplexNumber();

            c = sc.Subtraction(a, b);

            realans.Text = c.real.ToString();
            imagans.Text = c.imaginary.ToString();
            }
            }
      }
```

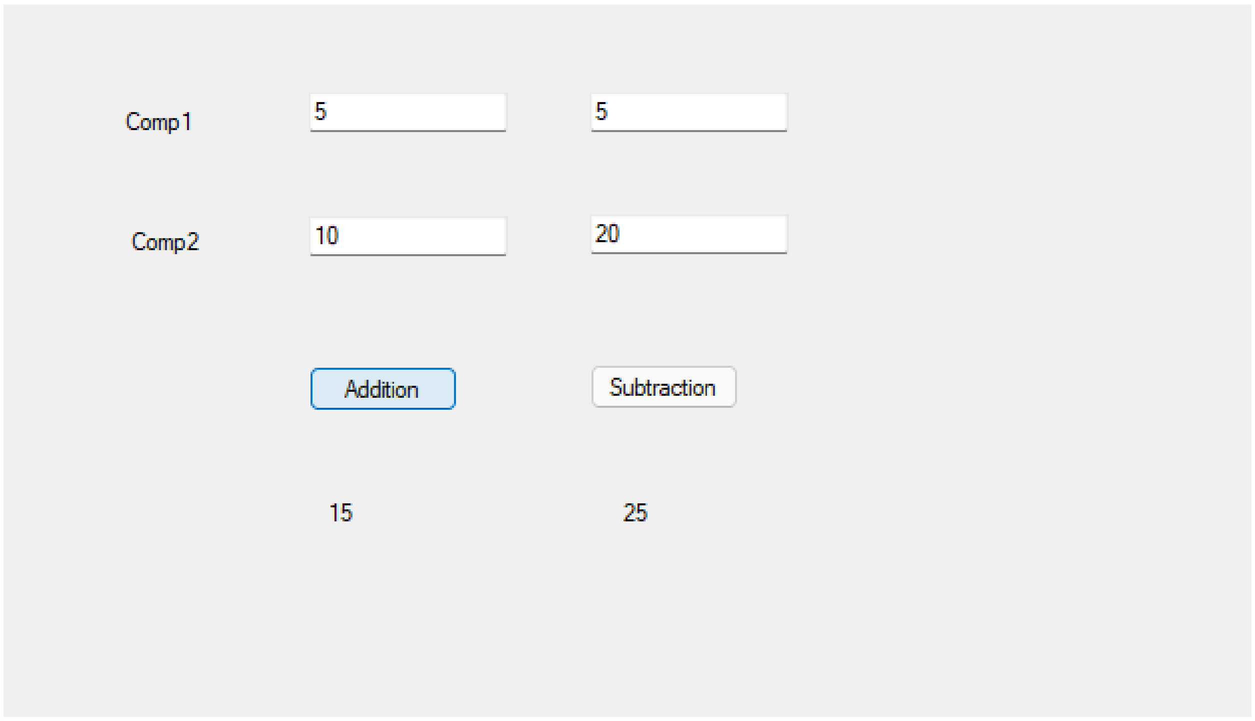
Client config file :

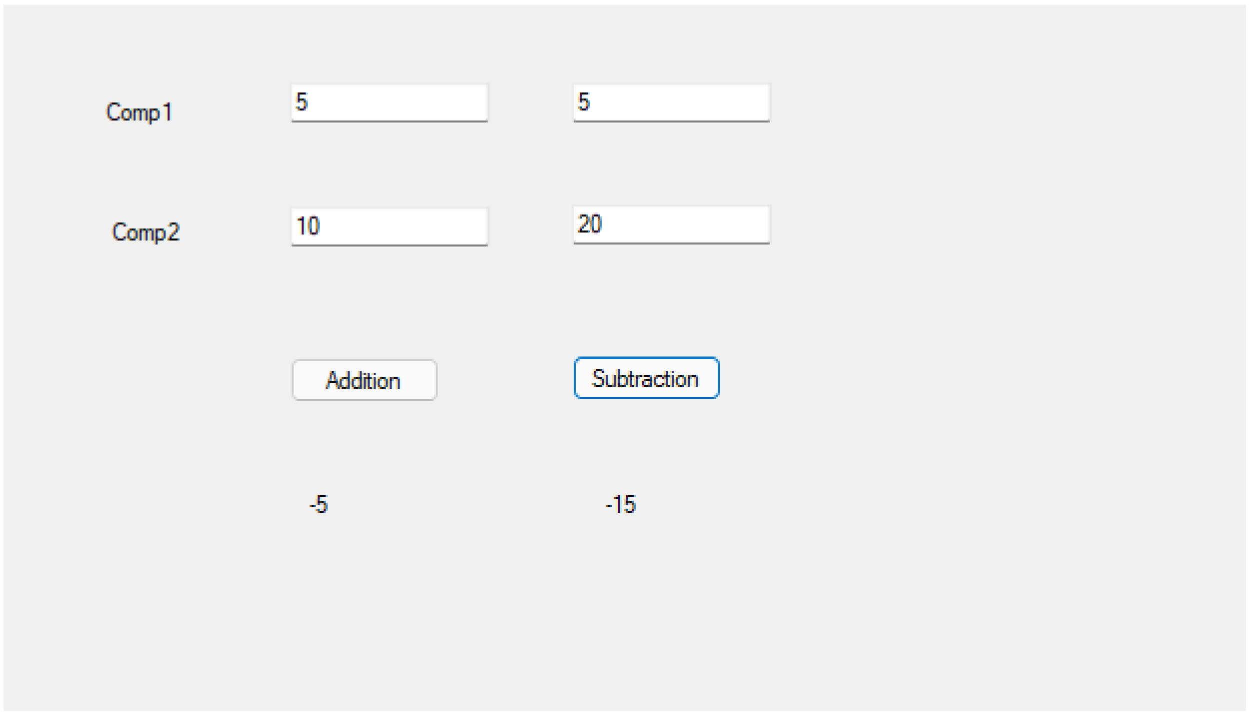
```
      <?xml version="1.0" encoding="utf-8" ?>
      <configuration>
            <startup>
            <supportedRuntime version="v4.0"
sku=".NETFramework,Version=v4.7.2" />
            </startup>
            <system.serviceModel>
            <bindings>
            <basicHttpBinding>
```

```
                <binding name="BasicHttpBinding_IService1" />
            </basicHttpBinding>
            </bindings>
            <client>
            <endpoint
address="http://localhost:8733/Design_Time_Addresses/WcfData
ContractCalc/Service1/"
                binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_IService1"
                contract="ServiceReference1.IService1"
name="BasicHttpBinding_IService1" />
            </client>
            </system.serviceModel>
        </configuration>
```

Output :




Comp1
5
5
Comp2
10
20
Addition
Subtraction
15
25


Comp1
5
5
Comp2
10
20
Addition
Subtraction
-5
-15

2. **Create a WCF Service to demonstrate the use of Data Contract with database connectivity for any data entity of your choice (employee, student etc.)**

   IEmployee :

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

namespace WCFDB
{

    [ServiceContract]
    public interface IEmployee
    {
        [OperationContract]
        DataSet GetAllEmp();

        [OperationContract]
        Employee GetEmpById(int id);



    }

    [DataContract]
    public class Employee
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public string Name { get; set; }
        [DataMember]
        public string Designation { get; set; }
    }
}
```

EmpService :

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

namespace WCFDB
{
    // 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 EmpService : IEmployee
    {
        public string GetData(int value)
        {
            return string.Format("You entered: {0}", value);
        }

        private string connectionString = "Data
Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=empdb;Integrated Security=True;Connect Timeout=30;";
        public DataSet GetAllEmp()
        {
            DataSet ds = new DataSet();
            string query = "select * from emp_info";
            SqlConnection con = new
SqlConnection(connectionString);
            using (con)
            {
                SqlDataAdapter dataAdapter = new
SqlDataAdapter(query, con);
                con.Open();

                dataAdapter.Fill(ds, "Employees");

                return ds;
            }
        }

        Employee IEmployee.GetEmpById(int id)
        {
            SqlConnection con = new
SqlConnection(connectionString);
```

```
                string query = "select * from emp_info where Id =
@EmployeeId";
                SqlDataAdapter sqlDataAdapter = new
SqlDataAdapter(query, con);

sqlDataAdapter.SelectCommand.Parameters.AddWithValue("@E
mployeeId", id);
                DataSet ds = new DataSet();
                con.Open();

                sqlDataAdapter.Fill(ds, "Employee");
                var row = ds.Tables["Employee"].Rows[0];

                Employee employee = new Employee();
                employee.Id = Convert.ToInt32(row["Id"]);
                employee.Name = Convert.ToString(row["Name"]);
                employee.Designation =
Convert.ToString(row["Designation"]);

                return employee;

            }
        }
    }
```

App.config :

```
<?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="WCFDB.EmpService">
              <endpoint address="" binding="basicHttpBinding"
contract="WCFDB.IEmployee">
                <identity>
                  <dns value="localhost" />
                </identity>
              </endpoint>
              <endpoint address="mex" binding="mexHttpBinding"
contract="IMetadataExchange" />
              <host>
                <baseAddresses>
                  <add
baseAddress="http://localhost:8733/Design_Time_Addresses/WC
FDB/Service1/" />
                </baseAddresses>
              </host>
```

```
    </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 code :

Webform1.aspx :

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

    <!DOCTYPE html>
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head runat="server">
      <title>Employee Information</title>
      <style>
        table {
          width: 100%;
          border-collapse: collapse;
        }
        th, td {
          border: 1px solid black;
          padding: 8px;
          text-align: left;
        }
      </style>
    </head>
    <body>
      <form id="form1" runat="server">
        <h2>Employee Information</h2>

        <!-- GridView to display employees -->
        <asp:GridView ID="EmployeeGridView" runat="server"
AutoGenerateColumns="True" Width="100%" />

        <br />
```

```
            <!-- Button to fetch all employees -->
            <asp:Button ID="button1" runat="server" Text="Get All
Employees" OnClick="button1_click" />

            <br /><br />

            <!-- TextBox to enter employee ID -->
            <asp:TextBox ID="textbox1" runat="server"
Placeholder="Enter Employee ID" />

            <!-- Button to get employee by ID -->
            <asp:Button ID="button2" runat="server" Text="Get
Employee By ID" OnClick="button2_click" />

            <br /><br />

            <!-- Label to show employee details by ID -->
            <asp:Label ID="lable1" runat="server" Text=""
ForeColor="Green" />

         </form>
      </body>
      </html>
```

webform1.aspx.cs :

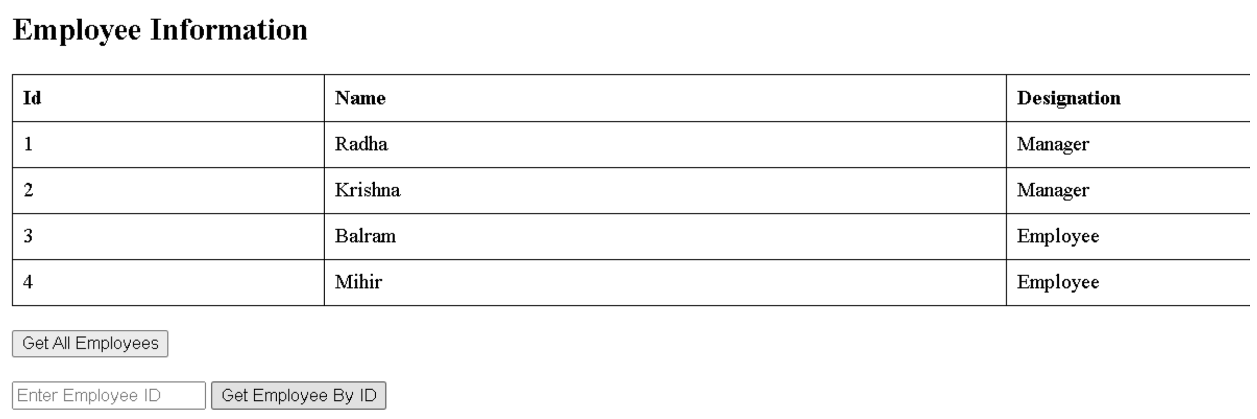
```
using ClientDB.ServiceReference1;
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Reflection.Emit;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace ClientDB
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void button1_click(object sender, EventArgs
e)
        {
            ServiceReference1.EmployeeClient ec = new
ServiceReference1.EmployeeClient();
            DataSet ds = new DataSet();
            ds = ec.GetAllEmp();
            EmployeeGridView.DataSource = ds;
            EmployeeGridView.DataBind();
        }

        protected void button2_click(object sender, EventArgs
e)
        {
            ServiceReference1.EmployeeClient ec = new
ServiceReference1.EmployeeClient();
            ServiceReference1.Employee emp =
ec.GetEmpById(Int32.Parse(textbox1.Text));
            lable1.Text = emp.Name;

        }
    }
}
```

Web.config :

```
    <?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="4.8" />
        <httpRuntime targetFramework="4.8" />
      </system.web>
      <system.codedom>
        <compilers>
          <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CS
harpCodeProvider,
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.VB
CodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform,
Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008
/define:_MYTYPE=\&quot;Web\&quot; /optionInfer+" />
            </compilers>
          </system.codedom>
          <system.serviceModel>
            <bindings>
              <basicHttpBinding>
                <binding name="BasicHttpBinding_IEmployee" />
              </basicHttpBinding>
            </bindings>
            <client>
              <endpoint
address="http://localhost:8733/Design_Time_Addresses/WCFDB/
Service1/"
                binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_IEmployee"
                contract="ServiceReference1.IEmployee"
name="BasicHttpBinding_IEmployee" />
            </client>
          </system.serviceModel>
```

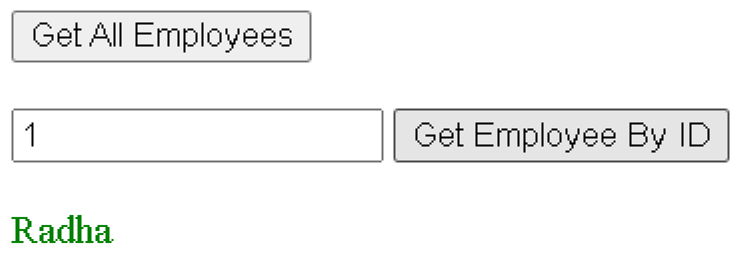
```
</configuration>
```

Output :



Employee Information

Id	Name	Designation
1	Radha	Manager
2	Krishna	Manager
3	Balram	Employee
4	Mihir	Employee



Radha