

Name :- Nisarg .k. Amlani

Roll :- CE001

Id :- 22ceueg082

Lab :- 5

1. Complex Number Data Contact Code and its config files

=> Host Config file

```
<?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="Lab_5_DataContract.Service1">
        <host>
          <baseAddresses>
            <add baseAddress =
"http://localhost:8733/Design_Time_Addresses/Lab_5_DataContract/Service1/" />
          </baseAddresses>
        </host>
        <!-- Service Endpoints -->
        <!-- Unless fully qualified, address is relative to base address supplied above -->
```

```
<endpoint address="" binding="basicHttpBinding"
contract="Lab_5_DataContract.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>
```

=> Interface Code

```
namespace Lab_5_DataContract
{
    [ServiceContract]
    internal interface IService1
    {
        [OperationContract]
        ComplexNumber Add(ComplexNumber number1, ComplexNumber number2);

        [OperationContract]
        ComplexNumber Subtract (ComplexNumber number1, ComplexNumber
number2);
    }
}
```

=> Interface Implementation

```
namespace Lab_5_DataContract
{
    // 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 ComplexNumber Add(ComplexNumber number1, ComplexNumber
number2)
        {
            ComplexNumber result = new ComplexNumber();
            result.Real = number1.Real + number2.Real;
            result.Imaginary = number1.Imaginary + number2.Imaginary;

            return result;
        }

        public ComplexNumber Subtract(ComplexNumber number1, ComplexNumber
number2)
        {
            ComplexNumber result = new ComplexNumber();
```

```

        result.Real = number1.Real - number2.Real;
        result.Imaginary = number1.Imaginary - number2.Imaginary;
        return result;
    }
}

```

=> Class Code

```

namespace Lab_5_DataContract
{
    [DataContract]
    public class ComplexNumber
    {
        [DataMember]
        public double Real { get; set; }
        [DataMember]
        public double Imaginary { get; set; }
    }
}

```

=> 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/Lab_5_DataContract/Service1/"
        binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_IService1"
        contract="ServiceReference1.IService1"
name="BasicHttpBinding_IService1" />
    </client>
</system.serviceModel>
</configuration>

```

=> Client Form Code

```

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

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void add_Click(object sender, EventArgs e)
        {

            ServiceReference1.Service1Client sc = new ServiceReference1.Service1Client();
            ServiceReference1.ComplexNumber cn1 = new
ServiceReference1.ComplexNumber();
            ServiceReference1.ComplexNumber cn2 = new
ServiceReference1.ComplexNumber();
            ServiceReference1.ComplexNumber res = new
ServiceReference1.ComplexNumber();

            cn1.Real = Double.Parse(tbr1.Text);
            cn1.Imaginary = Double.Parse(tbi1.Text);

```

```
cn2.Real = Double.Parse(tbr2.Text);
cn2.Imaginary = Double.Parse(tbi2.Text);
res = sc.Add(cn1, cn2);
```

```
reslb.Text = string.Format("The result is real {0} imaginary {1}
",res.Real,res.Imaginary);
```

```
}
```

```
private void sub_Click(object sender, EventArgs e)
{
    ServiceReference1.Service1Client sc = new ServiceReference1.Service1Client();
    ServiceReference1.ComplexNumber cn1 = new
ServiceReference1.ComplexNumber();
    ServiceReference1.ComplexNumber cn2 = new
ServiceReference1.ComplexNumber();
    ServiceReference1.ComplexNumber res = new
ServiceReference1.ComplexNumber();
```

```
cn1.Real = Double.Parse(tbr1.Text);
cn1.Imaginary = Double.Parse(tbi1.Text);
cn2.Real = Double.Parse(tbr2.Text);
cn2.Imaginary = Double.Parse(tbi2.Text);
res = sc.Subtract(cn1, cn2);
```

```
reslb.Text = string.Format("The result is real {0} imaginary {1} ", res.Real,
res.Imaginary);
```

```
}
```

```
}
```

```
}
```

=> Output Addition

A screenshot of a Windows application window titled "Form1". The window has a light gray background and standard Windows window controls (minimize, maximize, close) in the top right corner. The form contains four text input fields arranged in a 2x2 grid. The top row has "Real1" with the value "1" and "Imaginary1" with the value "2". The bottom row has "Real2" with the value "2" and "Imaginary2" with the value "4". To the right of these inputs are two buttons: "add" (highlighted with a blue dotted border) and "sub". Below the buttons, the text "The result is real 3 imaginary 6" is displayed.

Input	Value
Real1	1
Imaginary1	2
Real2	2
Imaginary2	4

Buttons: add, sub

Output: The result is real 3 imaginary 6

=> Output Substraction

A screenshot of a Windows application window titled "Form1". The window has a light gray background and standard Windows window controls (minimize, maximize, close) in the top right corner. The form contains four text input fields arranged in a 2x2 grid. The top row has "Real1" with the value "1" and "Imaginary1" with the value "2". The bottom row has "Real2" with the value "2" and "Imaginary2" with the value "4". To the right of these inputs are two buttons: "add" and "sub" (highlighted with a blue dotted border). Below the buttons, the text "The result is real -1 imaginary -2" is displayed.

Input	Value
Real1	1
Imaginary1	2
Real2	2
Imaginary2	4

Buttons: add, sub

Output: The result is real -1 imaginary -2

2. WCF Database Demo Service

=> Soap Message

GetAllEmp() :

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://tempu
      ri.org
      /IEmployee/GetAllEmp</Action>
    </s:Header>
    <s:Body>
      <GetAllEmp xmlns="http://tempuri.org/" />
    </s:Body>
  </s:Envelope>
```

Response :

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetAllEmpResponse xmlns="http://tempuri.org/">
      <GetAllEmpResult
        xmlns:a="http://schemas.datacontract.org/2004/07/WCF_DatabaseDemo_Service"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Employee>
          <a:Description>Manager</a:Description>
          <a:Id>1</a:Id>
```



```

<a:Name>Rahul</a:Name>
</a:Employee>
<a:Employee>
<a:Description>HR</a:Description>
<a:Id>2</a:Id>
<a:Name>Rishi</a:Name>
</a:Employee>
<a:Employee>
<a:Description>Tech Lead</a:Description>
<a:Id>3</a:Id>
<a:Name>Pankaj</a:Name>
</a:Employee>
<a:Employee>
<a:Description>CTO</a:Description>
<a:Id>4</a:Id>
<a:Name>Jay</a:Name>
</a:Employee>
</GetAllEmpResult>
</GetAllEmpResponse>
</s:Body>
</s:Envelope>

```

=> GetEmpById()

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://tempu
ri.org
/IEmployee/GetEmpById</Action>
</s:Header>
<s:Body>
<GetEmpById xmlns="http://tempuri.org/">
<id>1</id>

```

```
</GetEmpById>
</s:Body>
</s:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <GetEmpByIdResponse xmlns="http://tempuri.org/">
      <GetEmpByIdResult
        xmlns:a="http://schemas.datacontract.org/2004/07/WCF_DatabaseDemo_
        Service"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Description>Manager</a:Description>
        <a:Id>1</a:Id>
        <a:Name>Rahul</a:Name>
      </GetEmpByIdResult>
    </GetEmpByIdResponse>
  </s:Body>
</s:Envelope>
```

=> App.Config Code

```
<?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="Data_Contract.EmpService">
```

```
<host>
```

```
<baseAddresses>
```

```
<add baseAddress =
```

```
"http://localhost:8733/Design_Time_Addresses/Data_Contract/EmpService/" />
```

```
</baseAddresses>
```

```
</host>
```

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

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

-->

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

```
contract="Data_Contract.IEmployee">
```

```
<!--
```

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>

```

=> Service Interface

```

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

namespace Data_Contract
{
    [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; }
    }
}

```

=> Service Class

```

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

namespace Data_Contract
{
    // 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
    {
        private string connectionString = "Data Source=(localdb)\\mssqllocaldb;Initial
        Catalog=empdb2;Integrated Security=True;Pooling=False;";

        DataSet IEmployee.GetAllEmp()
        {
            DataSet dataSet = new DataSet();
            string query = "SELECT * from emp_info";
            using(SqlConnection connection = new SqlConnection(connectionString))
            {
                SqlDataAdapter dataAdapter = new SqlDataAdapter(query,
connection);
                connection.Open();
                dataAdapter.Fill(dataSet, "Employees");
            }

            return dataSet;
        }
    }
}

```

```

Employee IEmployee.GetEmpbyID(int id)
{
    SqlConnection connection = new SqlConnection(connectionString);
    string query = "select * from emp_info where Id=@EmployeeID";
    SqlDataAdapter dataAdapter = new SqlDataAdapter(query, connection);
    dataAdapter.SelectCommand.Parameters.AddWithValue("@EmployeeID",
id);
    DataSet dataset = new DataSet();
    connection.Open();
    dataAdapter.Fill(dataset, "Employee");
    var row = dataset.Tables["Employee"].Rows[0];
    Employee emp = new Employee();
    emp.Id = Convert.ToInt32(row["Id"]);
    emp.Name = row["Name"].ToString();
    emp.Designation = row["Designation"].ToString();
    return emp;
}
}
}

```

Client Code

=> Client App.Config Code

```

<?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.CSharpCodeProvi
der, 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_IEmployee" />
        </basicHttpBinding>
    </bindings>
    <client>
        <endpoint
address="http://localhost:8733/Design_Time_Addresses/Data_Contract/EmpServ
ice/"
        binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_IEmployee"
        contract="ServiceReference1.IEmployee"
name="BasicHttpBinding_IEmployee" />
    </client>
</system.serviceModel>
</configuration>

```

=> Client Form Code

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

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

        protected void GetAll_Click(object sender, EventArgs e)
        {
            ServiceReference1.EmployeeClient employeeClient =
                new ServiceReference1.EmployeeClient();
            DataSet ds = new DataSet();
            ds = employeeClient.GetAllEmp();
            GridView1.DataSource = ds.Tables[0];

            GridView1.DataBind();
        }

        protected void GetNameByID_Click(object sender, EventArgs e)
        {
            ServiceReference1.EmployeeClient employeeClient =
                new ServiceReference1.EmployeeClient();

            ServiceReference1.EmployeeClient employeeClient1 = new
            ServiceReference1.EmployeeClient();

            ServiceReference1.Employee emp =
            employeeClient1.GetEmpbyID(Int32.Parse(TB.Text));

            lbl.Text = emp.Name;
        }
    }
}

```

=> webform.aspx code


```

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Button ID="GetAll" runat="server" OnClick="GetAll_Click" Text="Get
All" />
            <br />
            <br />
            <br />
            <asp:GridView ID="GridView1" runat="server">
</asp:GridView>
            <br />
            <br />
            <asp:TextBox ID="TB" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Button ID="GetNameByID" runat="server"
OnClick="GetNameByID_Click" Text="GetNameByid" />
            <br />
            <br />
            <asp:Label ID="lbl" runat="server" Text="Name : label"></asp:Label>
            <br />
        </div>
    </form>
</body>
</html>

```

Form1

Id Of Employee : 2

Output :

	Id	Name	Description	
▶	2	Rishi	HR	

Get Employee

Employee List

Output :

	Id	Name	Description	
▶	1	Rahul	Manager	
	2	Rishi	HR	
	3	Pankaj	Tech Lead	
	4	Jay	CTO	

Get All Employee