Practical no 1

AIM: Write a program to implement to create a simple web service that converts the temperature from Fahrenheit to Celsius and vice versa.

Theory

Steps:

1. Create a Web Application of ”ASP.NET” using(.Net Framework 4.7).
2. Give a suitable title to the project and solution.
3. Initialize it as ”Empty” solution.
4. On Solution Explorer, right click and add a ”Web Service(asmx)” to the solution.
5. Moving ahead we already have the files initialized for returning ”Hello World”.
6. Remove/overwrite the ”Hello World” ”WebMethod” and add your own ”WebMethods” to the source file.
7. Save it and try out the web-service using the play button to host the web service on a local ”IIS Express” server.
8. Once the web service successfully runs and gives the desired output in form of XML documents; We will Proceed towards making client side pages.
9. In solution explorer, right click on the connected services tab and click on ”add service reference” option.
10. Next discover the web service we just created and click on the service you created and click ok.
11. Now you have successfully connected the web service to the solution.
12. Add a new WebForm in the same solution and start designing the client side UI.
13. After completing the UI design open the backend C# code.
14. Define methods for Button Clicks either explicitly or by double clicking the respective buttons in the design section of the form .
15. In the button click methods first create a SOAP object for the ”webservice.WebService1SoapClient()” class.

16. Using the SOAP object invoke the web-service methods and pass the appropriate data from the input field casted to the data type used in web-service method.
17. Next try running the ASPX page using the local server, add exception handling for handling exceptions if required.

Code:

WebService1.asmx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace Temp
{

    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]

    public class WebService1 : System.Web.Services.WebService
    {

        [WebMethod]
        public double celsius_to_farhenheit(double celsius)
        {
            return ((celsius * 9 / 5) + 32);
        }
        [WebMethod]
        public double farhenheit_to_celsius(double farhenheit)
        {
            return ((farhenheit - 32) * 5 / 9);

        }
    }
}
```

WebForm1.aspx:

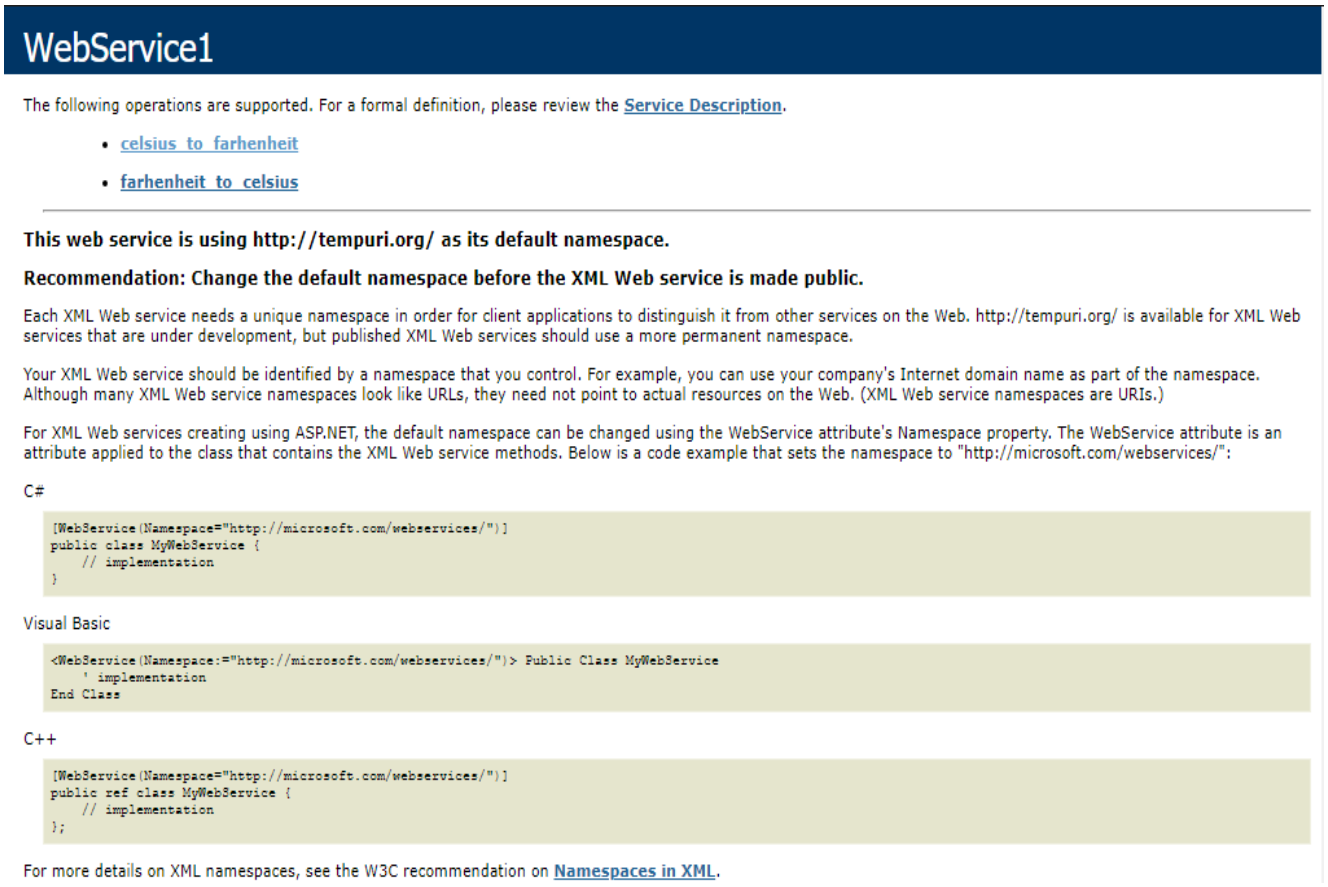
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Temp.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
<asp:Label ID="Label1" runat="server" Text="Input Temperature"></asp:Label>
              

            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>&deg;
            <asp:DropDownList ID="DropDownList1" runat="server">

                <asp:ListItem>Celsius</asp:ListItem>
                 <asp:ListItem>Farhenheit</asp:ListItem>

            </asp:DropDownList>

            <br/><br/>

            <asp:Button ID="Button1" runat="server" Text="Convert_to_Farhenhiet"
OnClick="Button1_Click"/>
                
            <asp:Button ID="Button2" runat="server" Text="Convert_to_Celsius"
OnClick="Button2_Click" />
            <br/><br/>

            <asp:Label ID="Label3" runat="server" Text="Result :"></asp:Label>
             
            <asp:Label ID="Label2" runat="server" Text=""></asp:Label>
             <asp:Label ID="Label4" runat="server" Text=""></asp:Label>
</div>
    </form>
</body>
</html>
```

WebForm1.aspx.cs

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

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

        }
        protected void Button1_Click(object sender, EventArgs e)
        {

            double result;
            try
            {
                ServiceReference1.WebService1SoapClient client = new
ServiceReference1.WebService1SoapClient();

                result = client.celsius_to_farhenheit(Convert.ToDouble(TextBox1.Text));

                if (DropDownList1.SelectedValue.Equals("Celsius"))
                {
                    Label2.Text = result.ToString();
                    Label4.Text = "&deg;Farhenheit";
                }
                if (DropDownList1.SelectedValue.Equals("Farhenheit"))
                {
                    Label2.Text = "Already in Farhenheit";
                    Label4.Text = "";
                }
            }
            catch (System.FormatException)
            {
                Label2.Text = "Invalid Inputs";
                Label4.Text = "";
            }

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            double result;
            try
            {
                ServiceReference1.WebService1SoapClient client = new
ServiceReference1.WebService1SoapClient();
                result = client.farhenheit_to_celsius((Convert.ToDouble(TextBox1.Text)));
                if (DropDownList1.SelectedValue.Equals("Celsius"))
```

```
                {
                    Label2.Text = "Already in Celsius";
                    Label4.Text = "";
                }
                if (DropDownList1.SelectedValue.Equals("Farhenheit"))
                {
                    Label2.Text = result.ToString();
                    Label4.Text = "&deg;celsius";
                }
            }
            catch (System.FormatException)
            {
                Label2.Text = "Invalid Inputs";
                Label4.Text = "";

            }
        }
    }
}
```

Outputs:



WebService1

The following operations are supported. For a formal definition, please review the Service Description.

- celsius_to_farhenheit
- farhenheit_to_celsius

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

For more details on XML namespaces, see the W3C recommendation on Namespaces in XML.

WebService1

Click here for a complete list of operations.

celsius_to_farhenheit

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
celsius:	

Invoke

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /WebService1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/celsius_to_farhenheit"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envel
  <soap:Body>
    <celsius_to_farhenheit xmlns="http://tempuri.org/">
      <celsius>double</celsius>
    </celsius_to_farhenheit>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envel
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<double xmlns="http://tempuri.org/">37.4</double>
```

Input Temperature 0 ° Farhenheit

Convert_to_Farhenhiet Convert_to_Celsius

Result : -17.7777777777778 °celsius

Input Temperature 3 ° Celsius

Convert_to_Farhenhiet Convert_to_Celsius

Result :

Input Temperature 3 ° Celsius

Convert_to_Farhenhiet Convert_to_Celsius

Result :