

Practical 17

Aim: Create a program for Web Service.

#PROGRAM

WebService1.asmx

```
<%@ WebService Language="C#" CodeBehind="WebService1.asmx.cs"  
Class="WebApplication3.WebService1" %>
```

WebService1.asmx.cs

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

```
namespace WebApplication3
```

```
{  
    /// <summary>  
    /// Summary description for WebService1  
    /// </summary>  
    [WebService(Namespace = "http://tempuri.org/")]  
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]  
    [System.ComponentModel.ToolboxItem(false)]  
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment  
the following line.  
    // [System.Web.Script.Services.ScriptService]  
    public class WebService1 : System.Web.Services.WebService  
    {  
  
        [WebMethod]  
        public int converttodaysweb(int day, int month, int year)  
        {  
            DateTime dt = new DateTime(year, month, day);          int datetodays =  
DateTime.Now.Subtract(dt).Days;          return datetodays;  
        }  
    }  
}
```

#OUTPUT

WebService1

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [converttodayweb](#)

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 created 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/web-services/":

C#

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

Visual Basic

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

C++

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

For more details on XML namespaces, see the W3C recommendation on [Namespaces in XML](#).

For more details on WSDL, see the [WSDL Specification](#).

For more details on URIs, see [RFC 2396](#).

WebService1

Click [here](#) for a complete list of operations.

converttodayweb

Test

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

Parameter	Value
day:	<input type="text" value="20"/>
month:	<input type="text" value="11"/>
year:	<input type="text" value="1996"/>

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.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/converttodayweb"

<?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/envelope/">
  <soap:Body>
    <converttodayweb xmlns="http://tempuri.org/">
      <day>int</day>
      <month>int</month>
      <year>int</year>
    </converttodayweb>
  </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/envelope/">
  <soap:Body>
```