

## Practical 1

tempervice.java:-

```
package mypack;
```

```
import javax.jws.WebService;
```

```
import javax.jws.WebMethod;
```

```
import javax.jws.WebParam;
```

```
@WebService(serviceName = "tempervice")
```

```
public class tempervice {
```

```
    @WebMethod(operationName = "hello")
```

```
    public String hello(@WebParam(name = "name") String txt) {
```

```
        return "Hello " + txt + " !";
```

```
    }
```

```
    @WebMethod(operationName = "F_to_C")
```

```
    public Double F_to_C(@WebParam(name = "val") double val) {
```

```
        return (val-32)*5/9;
```

```
    @WebMethod(operationName = "C_to_F")
```

```
    public Double C_to_F(@WebParam(name = "val") double val) {
```

```
        return (val*9/5)+32;
```

```
    }
```

```
}
```

---

c\_to\_f.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <%
      String d = request.getParameter("data");
      Integer dd = Integer.parseInt(d);
      try{
        mypack.Tempservice_Service service = new mypack.Tempservice_Service();
        mypack.Tempservice Port = service.getTempservicePort();
        double c = dd;

        java.lang.Double result = port.cToF(c);
        out.println("Result = "+result);
      }catch(Exception ex){

      }
    %>
  </body>
</html>
```

---

f\_to\_c.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <%
      String d = request.getParameter("data");
      Integer dd = Integer.parseInt(d);
      try{
        mypack.Tempservice_Service service = new mypack.Tempservice_Service();
        mypack.Tempservice Port = service.getTempservicePort();
        double f = dd;
        java.lang.Double result = port.fToC(f);
        out.println("Result = "+result);
      }catch(Exception ex){

      }
    %>
  </body>
</html>
```

---

index.html:

```
<html>
<head>
<title>todo Title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
</head>
<body>

<form>
<br><input type="text" name="data"><br>
<br><input type="submit" value="Convert F to C" name="ftoc" formaction="f_to_c.jsp"><br>
<br><input type="submit" value="Convert C to F" name="ctof" formaction="c_to_f.jsp"><br>

</form>

</body>
</html>
```

## Practical 2

index.html :=

```
<html>
<head>
<title> todo supply </title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
</head>
<body>
<form>
<h2>one way operation</h2><br>
<input type="text" name="ID" placeholder="Enter ID"><br><br>
<input type="submit" formaction="Del_data.jsp" value="Delete Data" > <br>
<h1>-----</h1>
<h2>Request-Response operation</h2><br><br>
<input type="submit" formaction="getData.jsp" value="Get Data">
</form>
</body>
</html>
```

-----

DeleteData.java:

```
package com.ss;
```

```
public class DeleteData{  
    public static void deldata(String id){  
        String a = id;  
        Data ob = new Data();  
        ob.remove(a);  
        System.out.println("Data is deleted.");  
    }  
}
```

-----

Del\_data.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
    <head>  
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
        <title>JSP Page</title>  
    </head>  
    <body>  
        <% page import="com.ss.DeleteData"%>  
        <%  
            String d = request.getParameter("ID");  
            DeleteData.deldata(id);  
        %>  
    </body>  
</html>
```

---

getData.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>JSP Page</title>


  <style>

table{

font-family:arial,sans-serif;

border-collapse:collapse;

}

td,th{

border:1px solid #000000;

text-align:center;

padding:0px;

}

</style>


  <script>

var request = new XMLHttpRequest();

request.open('GET','http://localhost:8080/Server/webresources/com.kk.friends/',true);

var data = JSON.parse(this.response);
```

```
for (var i = 0 ; i < data.length;i++){  
    var table = document.getElementById("myTable");  
    var row = table.insertRow();  
    var cell1 = row.insertCell(0);  
    var cell2 = row.insertCell(1);  
    cell1.innerHTML =data[i].id;  
    cell2.innerHTML =data[i].firstname;  
}  
};  
request.send();  
</script>  
</head>  
    <body>  
        <table id="myTable">  
            <tr>  
                <th> ID</th>  
                <th> NAME</th>  
            </tr>  
        </table>  
    </body>  
</html>
```



### Practical 3

CCipherService.java

```
package com.abc;
```

```
import java.util.Random;
```

```
import javax.jws.WebService;
```

```
import javax.jws.WebMethod;
```

```
import javax.jws.WebParam;
```

```
@WebService(serviceName = "CCipherService")
```

```
public class CCipherService{
```

```
@WebMethod(operationName="encrypt")
```

```
public String encrypt(@WebParam(name="name") String msg){
```

```
String ct="";
```

```
Random rd=new Random();
```

```
for(int i=0; i<msg.length(); i++) {
```

```
ct+=(char) (msg.charAt(i)+rd.nextInt(26));
```

```
}
```

```
return ct;
```

```
}
```

```
}
```

---

CCipherClient.java :

```
package ccipherclient;
```

```
import java.util.Scanner;
```

```
public class CCiperClient{  
    public static void main(String[] args){  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter message");  
        String msg=sc.next();  
        System.out.println("Encrypted message: "+encrypt(msg));  
    }  
}
```

```
private static String encrypt(java.lang.String name){  
    com.abc.CCipherService_Service = new com.abc.CCipherService_Service();  
    com.abc.CCipherService port = service.getCCipherServicePort();  
    return port.encrypt(name);  
}}
```

---

## Practical 4

Factorialwebservice.java:

```
package abc.com;
```

```
import javax.jws.WebService;
```

```
import javax.jws.WebMethod;
```

```
import javax.jws.WebParam;
```

```
@WebService(serviceName = "Factorialwebservice")
```

```
public class Factorialwebservice{
```

```
@WebMethod(operationName = "factorial")
```

```
public double factorial(@WebParam(name = "num") double num){
```

```
double f=1;
```

```
int i;
```

```
for(i=1;i<=num;i++){
```

```
f = f*i;
```

```
}
```

```
return f;
```

```
}
```

```
}
```

---

WebService\_Practical4\_Client:-

space WebService\_Practical4\_Client

```
public partial class webServicePractical4_Client : System.Web.UI.Page{
```

```
protected void Page_Load(object sender,EventArgs e){
```

```
}
```

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

```
ServiceReference1.FactorialwebserviceClient client = new  
ServiceReference1.FactorialwebserviceClient();
```

```
Label1.Text = "Factorial is: "+ client.factorial(Convert.ToDouble(TextBox1.Text)).ToString();
```

```
}}
```

---

## Practical 5

calcservice.java:

```
package mypack;
```

```
import javax.jws.WebService;
```

```
import javax.jws.WebMethod;
```

```
import javax.jws.WebParam;
```

```
@WebService(serviceName = "calcservice")
```

```
public class calcservice{
```

```
    @WebMethod(operationName = "Addition")
```

```
    public Integer Additon(@WebParam(name = "a") int a , @WebParam(name = "b") int b){
```

```
        return a+b;
```

```
    }
```

```
    @WebMethod(operationName = "Subtraction")
```

```
    public Integer Subtraction(@WebParam(name = "a") int a , @WebParam(name = "b") int b){
```

```
        return a-b;
```

```
    }
```

---

index.html:

```
<html>
```

```
<head>
```

```
<title>todo Title</title>
```

```
<meta charset="UTF-8">
<meta name="viewport" content=with=device-width,initial-scale=1.0">
</head>
<body>

<form action="index.jsp">
Enter n1<input type="text" name="n1">
Enter n2<input type="text" name="n2">
<input type="submit" value="submit">
</form>

</body>
</html>
```

---

index.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1> hello world!</h1>

    <%
```

```
int x = Integer.parseInt(request.getParameter("n1"));
```

```

int y = Integer.parseInt(request.getParameter("n2"));

try{
    mypack.Calcservice_Service service = new mypack.Calcservice_Service();
    mypack.Calcservice port = service.getCalcservicePort();

    int a = x;
    int b = y;
    java.lang.Integer result = port.addition(a,b);
    out.println("Result = "+result);
}catch(Exception ex){

}

%>
</body>
</html>

```

---

index.jsp:

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1> hello world!</h1>

```

```
<%  
int m = Integer.parseInt(request.getParameter("n1"));  
int n = Integer.parseInt(request.getParameter("n2"));  
try{  
    mypack.Calcservice_Service service = new mypack.Calcservice_Service();  
    mypack.Calcservice port = service.getCalcservicePort();  
    int a = m;  
    int b = n;  
    java.lang.Integer result = port.Substraction(a,b);  
    out.println("Result = "+result);  
}catch(Exception ex){  
  
}  
%>  
</body>  
</html>
```



## Practical 6

index.html:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
```

```
    <title>JSP Page</title>
```

```
  <style>
```

```
    table{
```

```
      font-family:arial,sans-serif;
```

```
      border-collapse:collapse;
```

```
    }
```

```
    td,th{
```

```
      border:1px solid #000000;
```

```
      text-align:center;
```

```
      padding:8px;
```

```
    }
```

```
  </style>
```

```
  <script>
```

```
    var request = new XMLHttpRequest();
```

```
    request.open('GET','http://localhost:8080/Server/webresources/com.kk.friends/',true);
```

```
    request.onload = function(){
```

```
      var data = JSON.parse(this.response);
```

```
      for (var i = 0 ; i < data.length;i++){
```

```
var table = document.getElementById("myTable");
var row = table.insertRow();
var cell1 = row.insertCell(0);
var cell2 = row.insertCell(1);
cell1.innerHTML = data[i].id;
cell2.innerHTML = data[i].firstname;
}
};
request.send();
</script>
</head>
<body>
  <table id="myTable">
    <tr>
      <th> ID</th>
      <th> NAME</th>
    </tr>
  </table>
</body>
</html>
```

## Practical 8

MathService.svx.cs:

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

namespace MathService1
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name
    // "MathService" in code, svc and config file together.

    // NOTE: In order to launch WCF Test Client for testing this service, please select MathService.svc
    // or MathService.svc.cs at the Solution Explorer and start debugging.

    public class MathService : IMathService
    {
        public Int32 add(Int32 n1, Int32 n2)
        {
            return add(n1, n2);
        }

        public Int32 subtract(Int32 n1, Int32 n2)
        {
            return subtract(n1, n2);
        }
    }
}
```

```

    }

    public Int32 multiply(Int32 n1, Int32 n2)
    {
        return multiply(n1, n2);
    }

    public Int32 divide(Int32 n1, Int32 n2)
    {
        return divide(n1, n2);
    }
}

```

---

IMathService.cs

```

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

namespace MathService1
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface
    name "IMathService" in both code and config file together.

    [ServiceContract]
    public interface IMathService
    {

```

[OperationContract]

Int32 add(Int32 n1 , Int32 n2);

[OperationContract]

Int32 subtract(Int32 n1, Int32 n2);

[OperationContract]

Int32 multiply(Int32 n1, Int32 n2);

[OperationContract]

Int32 divide(Int32 n1, Int32 n2);

}

}

---

form1.cs:

using MathServiceTestApp1.MathService1;

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 MathServiceTestApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            MathServiceClient loclient = new MathServiceClient();
            Int32 num1 = Convert.ToInt32(textBox1.Text.Trim());
            Int32 num2 = Convert.ToInt32(textBox2.Text.Trim());
            if (comboBox1.Text == "add")
            {
                textBox3.Text = loclient.add(num1, num2).ToString();
            }
            else if (comboBox1.Text == "subtract")
            {
                textBox3.Text = loclient.subtract(num1, num2).ToString();
            }
            else if (comboBox1.Text == "multiply")
            {
                textBox3.Text = loclient.multiply(num1, num2).ToString();
            }
            else
            {
                textBox3.Text = loclient.divide(num1, num2).ToString();
            }
        }
    }
}
```

```
    }  
    }  
}
```

## Practical 9

GreetingService.cs :

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Runtime.Serialization;  
using System.ServiceModel;  
using System.ServiceModel.Activation;  
using System.ServiceModel.Web;  
using System.Text;  
  
namespace Website3  
{  
    [ServiceContract(Namespace = "")]  
    [AspNetCompatibilityRequirements(RequirementsMode =  
        AspNetCompatibilityRequirementsMode.Allowed)]  
    public class GreetingService  
    {  
        // To use HTTP GET, add [WebGet] attribute. (Default ResponseFormat is  
        WebMessageFormat.Json)  
        // To create an operation that returns XML,  
        // add [WebGet(ResponseFormat=WebMessageFormat.Xml)],  
        // and include the following line in the operation body:  
        // WebOperationContext.Current.OutgoingResponse.ContentType = "text/xml";  
        [OperationContract]
```

```

    public void DoWork()
    {
        // Add your operation implementation here

        return;
    }

    [OperationContract]
    public string Greeting()
    {
        return "hello";
    }

    // Add more operations here and mark them with [OperationContract]
}
}

```

---

Default.aspx:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="Website3.Default" %>

```

```

<!DOCTYPE html>

```

```

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

```

```

<head runat="server">

```

```

    <title></title>

```



```
</head>

<body>

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

    <div>


      <asp:ScriptManager ID="ScriptManager1" runat="server">

        <Services>

          <asp:ServiceReference Path="~/GreetingService.svc.cs"

        </Services>

      </asp:ScriptManager>

      <input type="button" value="greet" onclick="showgreeting()" />

    </div>

  </form>

</body>


</html>


<script language="javascript" type="text/javascript">

  function showgreeting() {

    GreetingService.Greeting(onSuccess, onError);

  }


  function onSuccess(response) {

    alert(response);

  }


  function onError(error) {

    alert("an error occurred" + error.get_message());

  }


</script>
```

## Practical 10

Project\_name : SampleCalcSvc

file\_name : ICalcService (add item Interface)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.ServiceModel;
using System.Text;
using System.Threading.Tasks;
```

```
namespace SampleCalcSvc
{
    [ServiceContract]
    internal interface ICalcService
    {
        [OperationContract]
        int GetSum(int a, int b);
    }
}
```

---

file\_name: CalcService(add item class)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace SampleCalcSvc
```

```
{
```

```
    internal class CalcService
```

```
    {
```

```
        public int GetSum(int a , int b)
```

```
        {
```

```
            return a + b;
```

```
        }
```

```
    }
```

```
}
```