**JAVA RMI**

Steps: Compile all programs. Run rmic tool as follows:

1. rmic ServerImple
2. start rmiregistry

Execute MainServer and then Client.

**Main Server.java**

import java.net.\*;

import java.rmi.\*;

public class MainServer

{

public static void main(String args[])

{

try{

ServerImple ob1 = new ServerImple();

Naming.rebind("RMI",ob1);

System.out.println("Server Started");

}

catch(Exception e)

{

System.out.println("Exception" +e);

}

}

}

**ServerInterface.java**

import java.rmi.\*;

public interface ServerInterface extends Remote

{

public double add(double d1,double d2) throws RemoteException;

public double sub(double d1,double d2) throws RemoteException;

public double mul(double d1,double d2) throws RemoteException;

public double div(double d1,double d2) throws RemoteException;

public double mod(double d1,double d2) throws RemoteException;

}

**ServerImple.java**

import java.rmi.\*;

import java.rmi.server.\*;

public class ServerImple extends UnicastRemoteObject implements ServerInterface

{

public ServerImple() throws RemoteException{

super();

}

public double add(double d1,double d2) throws RemoteException

{

return d1 + d2;

}

public double sub(double d1,double d2)

{

return(d1-d2);

}

public double mul(double d1,double d2)

{

return(d1\*d2);

}

public double div(double d1,double d2)

{

return(d1/d2);

}

public double mod(double d1,double d2)

{

return(d1%d2);

}

}

**Client.java**

import java.rmi.\*;

public class Client

{

public static void main(String args[])

{

try{

ServerInterface si = (ServerInterface)Naming.lookup("RMI");

System.out.println("Found the service");

System.out.println("Service selected" +args[0]);

System.out.println("The First number is" +args[1]);

double d1 = Double.valueOf(args[1]).doubleValue();

System.out.println("The Second number is" +args[2]);

double d2 = Double.valueOf(args[2]).doubleValue();

System.out.println("Add:\t"+si.add(d1,d2));

System.out.println("Sub:\t"+si.sub(d1,d2));

System.out.println("Mul:\t"+si.mul(d1,d2));

System.out.println("Div:\t"+si.div(d1,d2));

System.out.println("Mod:\t"+si.mod(d1,d2));

}

catch(Exception e){

System.out.println("Exception" +e);

}

}

}

**Celcius To Fahrenheit**

Steps: Create A Web Application in NetBeans. Create new Web Service , add the code. Right click on Web Application (Build,Deploy,Test). Right click Web Service and Test Web Service.

**WebService1.java**

package kiran;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

@WebService(serviceName = "WebService1")

public class WebService1 {

@WebMethod(operationName = "convertCtoF")

public String convertCtoF(@WebParam(name = "cel") double cel) {

return "Celsius: " + cel + "Fahrenheit: " + ((cel\*1.8)+32);

}

@WebMethod(operationName = "convertFtoC")

public String convertFtoC(@WebParam(name = "fah") double fah) {

return "Fahrenheit: " + fah + "Celsius: " + (((fah-32)\*5)/9);

}

}

**Product Listing**

Steps: Create A Web Application in NetBeans. Create new Web Service , add the code. Create another Web Service for Exception. Right click on Web Application (Build,Deploy,Test). Right click Web Service and Test Web Service.

**WebService2.java**

package kiran2;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import java.util.Hashtable;

@WebService(serviceName = "NewWebService")

public class WebService2 {

public Hashtable products;

public WebService2(){

products=new Hashtable();

products.put("Duffle Bag",new Double(495.90));

products.put("Sling Bag",new Double(459.01));

products.put("Sequil Bag",new Double(400.99));

products.put("Backpack",new Double(459.99));

products.put("School Bag",new Double(320.99));

products.put("Gym Bag",new Double(359.99));

}

@WebMethod(operationName = "GetPrice")

public double GetPrice(@WebParam(name="name")String txt) throws ProductNotFoundException {

Double price;

price=(Double)products.get(txt);

if(price==null)

{

throw new ProductNotFoundException("products"+products+"notfound");

}

else{

return price.doubleValue();

}

}

}

**ProductNotFoundException.java**

package kiran2;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import com.sun.tools.ws.processor.model.Fault;

public class ProductNotFoundException extends Exception {

public Fault fault;

ProductNotFoundException(String faultString,Fault fault){

super();

this.fault=fault;

}

ProductNotFoundException(String string) {

super();

this.fault=fault;

}

}

**Invoke a Calculator Web Service using Java Application Client**

Steps: Create A Web Application in NetBeans. Create new Web Service , add the code. Right click on Web Application (Build,Deploy,Test). Right click Web Service and Test Web Service.

Now Create A Java Application. Right Click and create new Web Service Client. Browse the project and set it to the Web Service that we created. Give it a package name and add the code. Run to get the output in the terminal.

**ServerCal.java (Web Service Server)**

package clientcalc.servercal;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

@WebService(serviceName = "ServerCal")

public class ServerCal {

@WebMethod(operationName = "add")

public Double add(@WebParam(name = "a")double a,@WebParam(name = "b")double b) {

return (a+b);

}

@WebMethod(operationName = "subract")

public Double sub(@WebParam(name = "a")double a,@WebParam(name = "b")double b) {

return (a-b);

}

@WebMethod(operationName = "multiply")

public Double mul(@WebParam(name = "a")double a,@WebParam(name = "b")double b) {

return (a\*b);

}

@WebMethod(operationName = "divide")

public Double div(@WebParam(name = "a")double a,@WebParam(name = "b")double b) {

return (a/b);

}

}

**CalcClient.java (Web Service Client)**

package calcclient;

public class CalcClient {

public static void main(String[] args) {

System.out.println("Addition: " + add(12, 22.45));

System.out.println("Subtraction: " + sub(12, 2));

System.out.println("Muliplication: " + mul(17, 45));

System.out.println("Division: " + div(33, 11));

}

private static Double add(double a, double b){

ServerCal\_Service service = new ServerCal\_Service();

ServerCal port = service.getServerCalPort();

return port.add(a, b);

}

private static Double sub(double a, double b){

ServerCal\_Service service = new ServerCal\_Service();

ServerCal port = service.getServerCalPort();

return port.subract(a, b);

}

private static Double mul(double a, double b){

ServerCal\_Service service = new ServerCal\_Service();

ServerCal port = service.getServerCalPort();

return port.multiply(a, b);

}

private static Double div(double a, double b){

ServerCal\_Service service = new ServerCal\_Service();

ServerCal port = service.getServerCalPort();

return port.divide(a, b);

}

}

**Web Service using Servlet**

Steps: Create a Web Application in NetBeans. Create a Web Service Client and give the location of our already created Web Service. Create a new Servlet page, add the code. Now make changes to the index.html page. Right click on Web Application (Build,Deploy,Test).Run the Web Application.

**CalcServlet.java**

package org.me.add;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.xml.ws.WebServiceRef;

@WebServlet(name = "CalcServlet", urlPatterns = {"/CalcServlet"})

public class CalcServlet extends HttpServlet {

@WebServiceRef(wsdlLocation = "http://localhost:8080/WebApplication3/ServerCal?WSDL")

public org.me.add.ServerCal\_Service ser1;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

ServerCal port = ser1.getServerCalPort();

double i = Double.parseDouble(request.getParameter("num1"));

double j = Double.parseDouble(request.getParameter("num2"));

double res = 0;

if(request.getParameter("button1")!=null)

{

res = port.add(i,j);

}

if(request.getParameter("button2")!=null)

{

res = port.subract(i,j);

}

if(request.getParameter("button3")!=null)

{

res = port.multiply(i,j);

}

if(request.getParameter("button4")!=null)

{

res = port.divide(i,j);

}

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet CalcServlet</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1> The result is equal to " + res + "</h1>");

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**Index.html**

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

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

</head>

<body>

<div>

<form action = "CalcServlet">

First Number <input type = "number" name = "num1"><br>

Second Number <input type = "number" name = "num2"><br>

<input type = "submit" name = "button1" value = "Add" onclick="CalcServlet">

<input type = "submit" name = "button2" value = "Subtract" onclick="CalcServlet">

<input type = "submit" name = "button3" value = "Multiply" onclick="CalcServlet">

<input type = "submit" name = "button4" value = "Divide" onclick="CalcServlet">

</form>

</div>

</body>

</html>

**Consume a Web Service using JSP page**

Steps: Create a Web Application in NetBeans. Delete index.html page. Create a Web Service Client and give the location of our already created Web Service. Create a new JSP page. Now add the operation (represented by a red dot) eg: “Add” by dragging it and dropping it below the h1 tag in the JSP page. Change the value of a and b to any random numbers. Right click on Web Application (Build,Deploy,Test). Run the Web Application.

* Self generated code ( For creating the Web Service refer to ServerCal.java)

**Creating Web Service using WCF**

Steps: New Project -> APS.NET Web Application -> Set name as StockService

Select Empty Web App

Next, Go to Solution Explorer and right click on Project name -> Add -> New Item -> Web Service

Rename it as StockService.asmx. Add the code.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

namespace Product

{

/// <summary>

/// Summary description for ProductWebService

/// </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 ProductWebService : System.Web.Services.WebService

{

string[,] stock = {

{ "REL", "Reliance", "999.45" },

{ "ICICI", "ICICI Corp.", "1051.2"},

{ "JIO", "Rel. Jio", "892.05" }

};

[WebMethod]

public double getPrice(string symbol)

{

double result = 0;

for (int i = 0; i< stock.GetLength(0) ;i++)

{

if (String.Compare(stock[0, i], symbol, true)==0)

{

result= Convert.ToDouble(stock[i, 2]);

}

}

return result;

}

[WebMethod]

public string getName(string symbol)

{

for (int i = 0; i < stock.GetLength(0); i++)

{

if (String.Compare(stock[i,0], symbol, true) == 0)

{

return (stock[i, 1]);

}

}

return "Not Found";

}

}

}

**Create a Web Service client using WCF**

Steps: New Project -> APS.NET Web Application -> Set name as StockClient

Select Empty Web App

Next, Go to Solution Explorer and right click on Project name -> Add -> New Item -> Reference

Now select the reference as the previously created Web Service.

Add 4 labels (2 for text and 2 for output) from the toolbox.

Add 2 buttons for calculating price and name of the stock. Also 2 textboxes for entering Stock symbols

