

Code First Approach for Creating SOAP Web Service

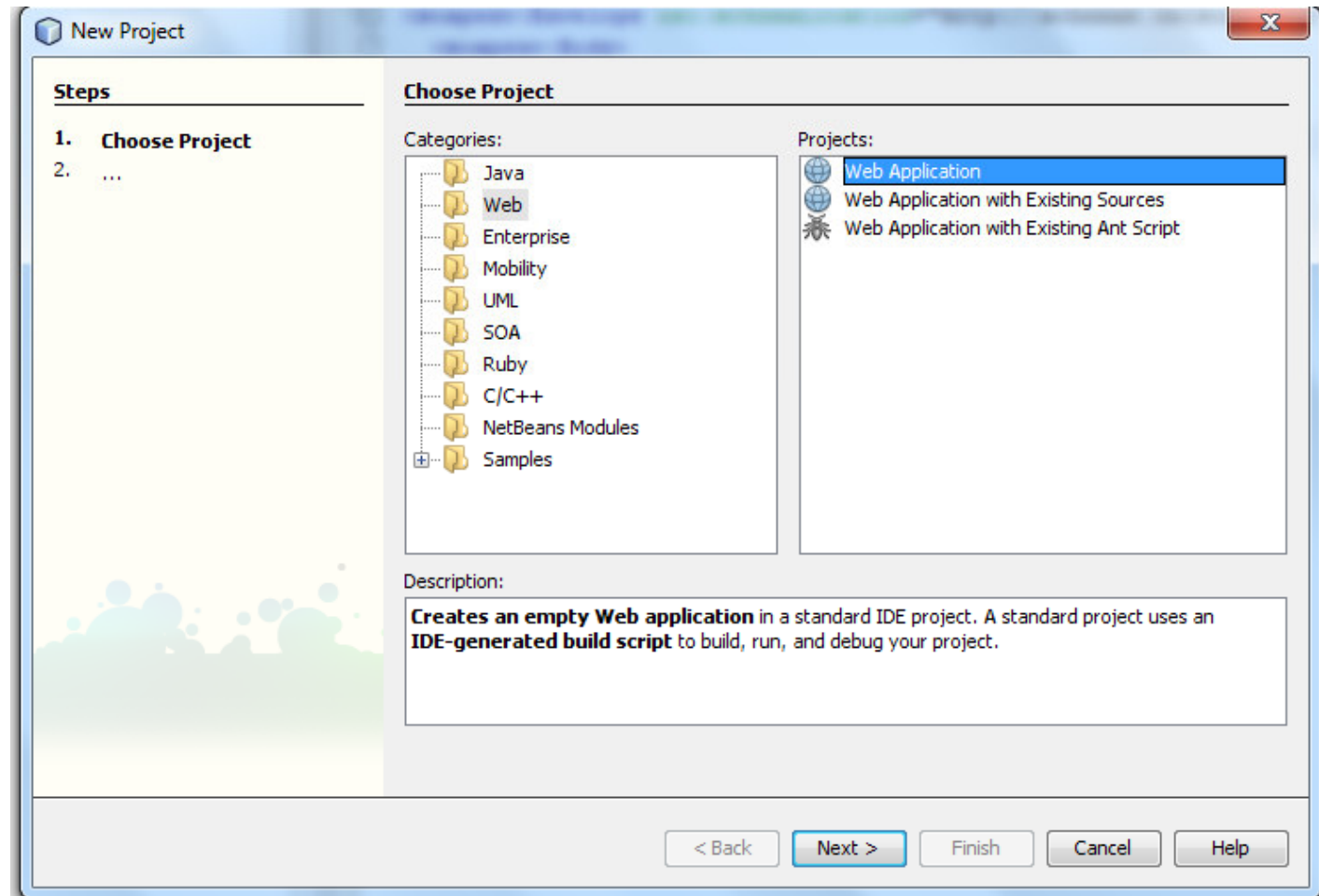
Prof. (Dr.) Vipul K. Dabhi
Assoc. Professor,
Department of Information Technology,
D. D. University

Approaches for creating SOAP web-service

- **Code First Approach**
 - Writing source code for web service and then WSDL File
- **Contract First Approach**
 - Creating Web-service based on given WSDL File

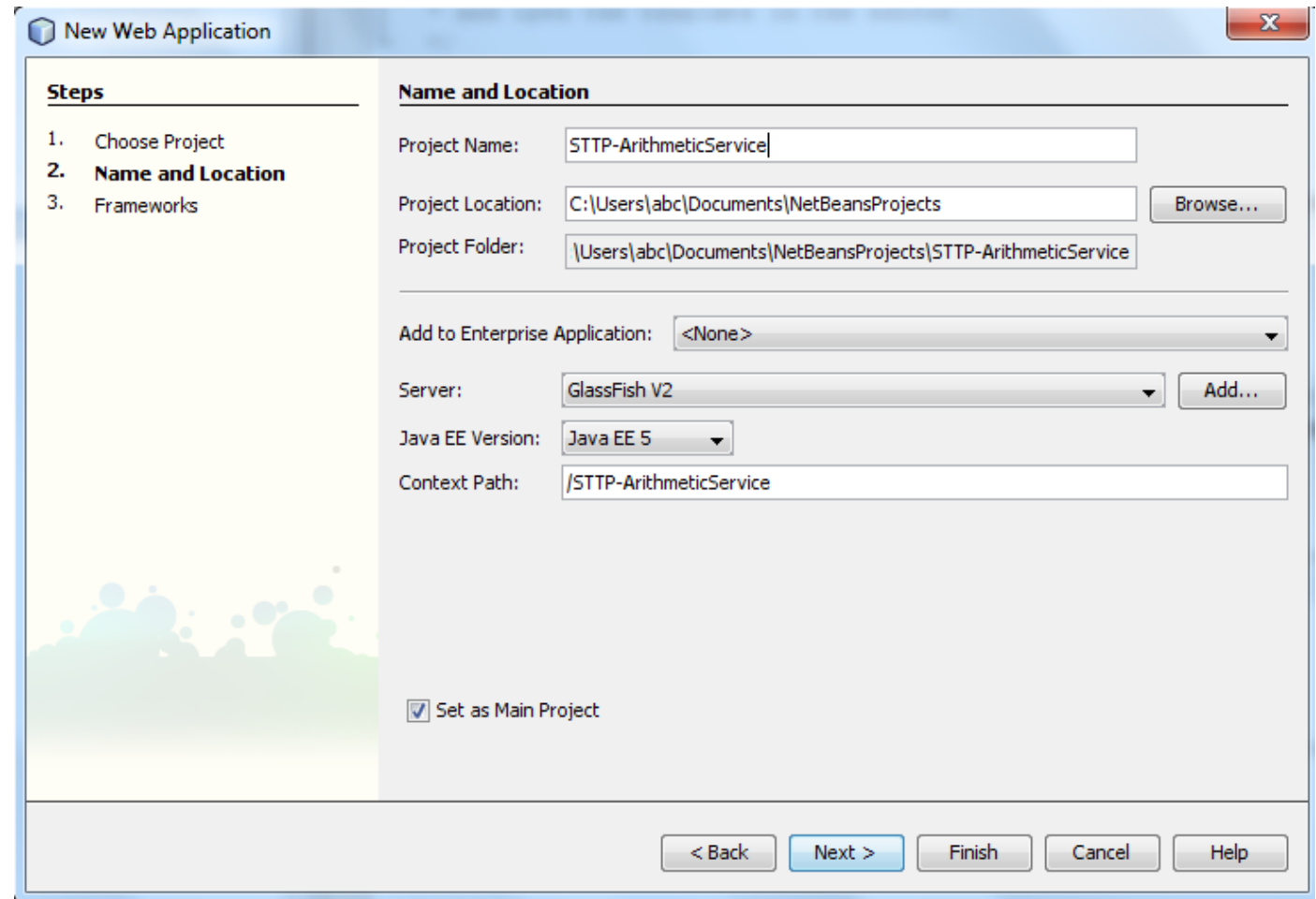
Creating Web Service using Code First Approach

- **Create Web Application**
 - Give it name Arithmetic



Creating Web Service using Code First Approach

- **Create Web Application**
- **Give it name STTP-ArithmeticService**



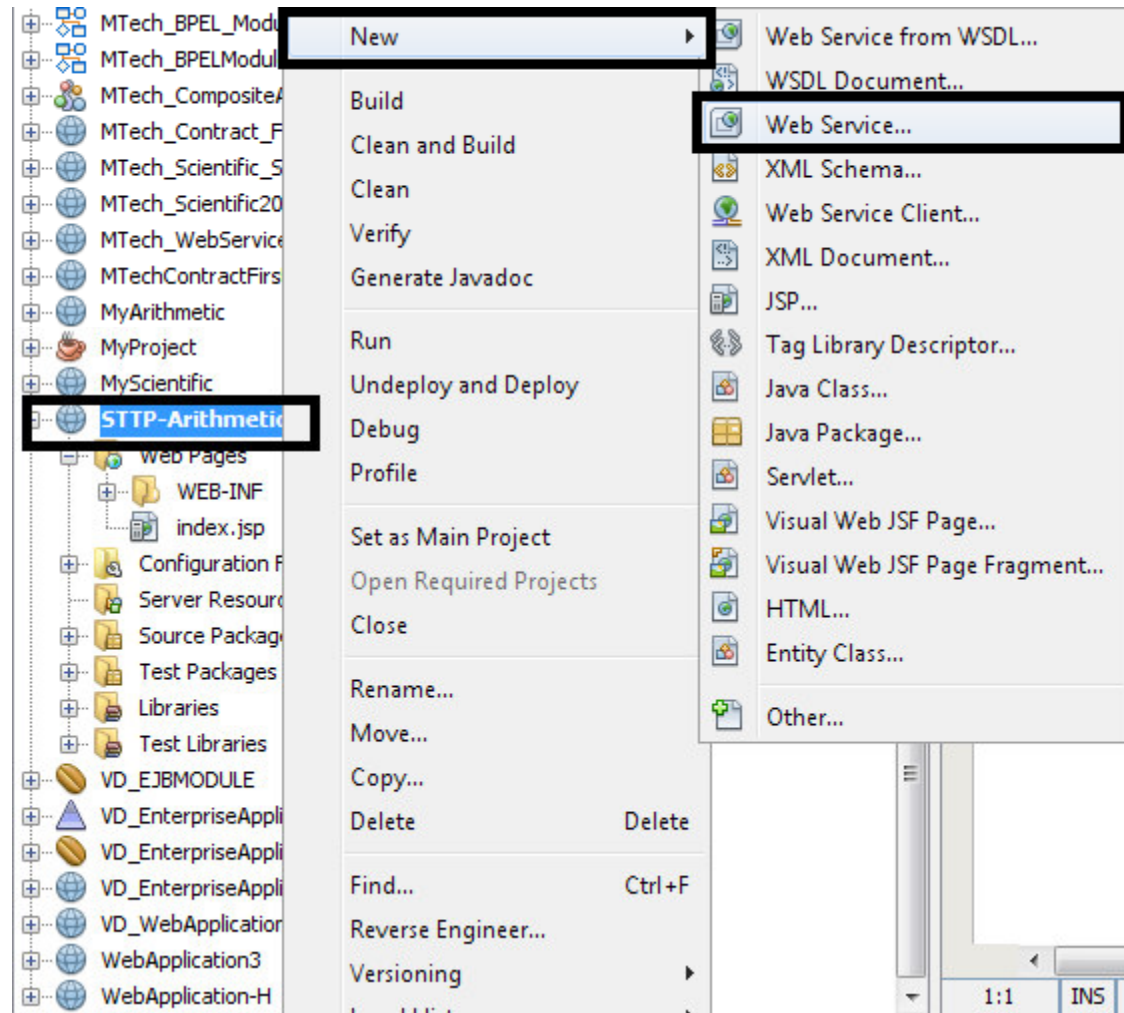
The screenshot shows the 'New Web Application' dialog box in NetBeans IDE. The 'Steps' panel on the left indicates the current step is '2. Name and Location'. The 'Name and Location' panel contains the following fields:

- Project Name:** STTP-ArithmeticService
- Project Location:** C:\Users\abc\Documents\NetBeansProjects (with a 'Browse...' button)
- Project Folder:** \Users\abc\Documents\NetBeansProjects\STTP-ArithmeticService
- Add to Enterprise Application:** <None>
- Server:** GlassFish V2 (with an 'Add...' button)
- Java EE Version:** Java EE 5
- Context Path:** /STTP-ArithmeticService
- ☒ **Set as Main Project**

At the bottom of the dialog are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

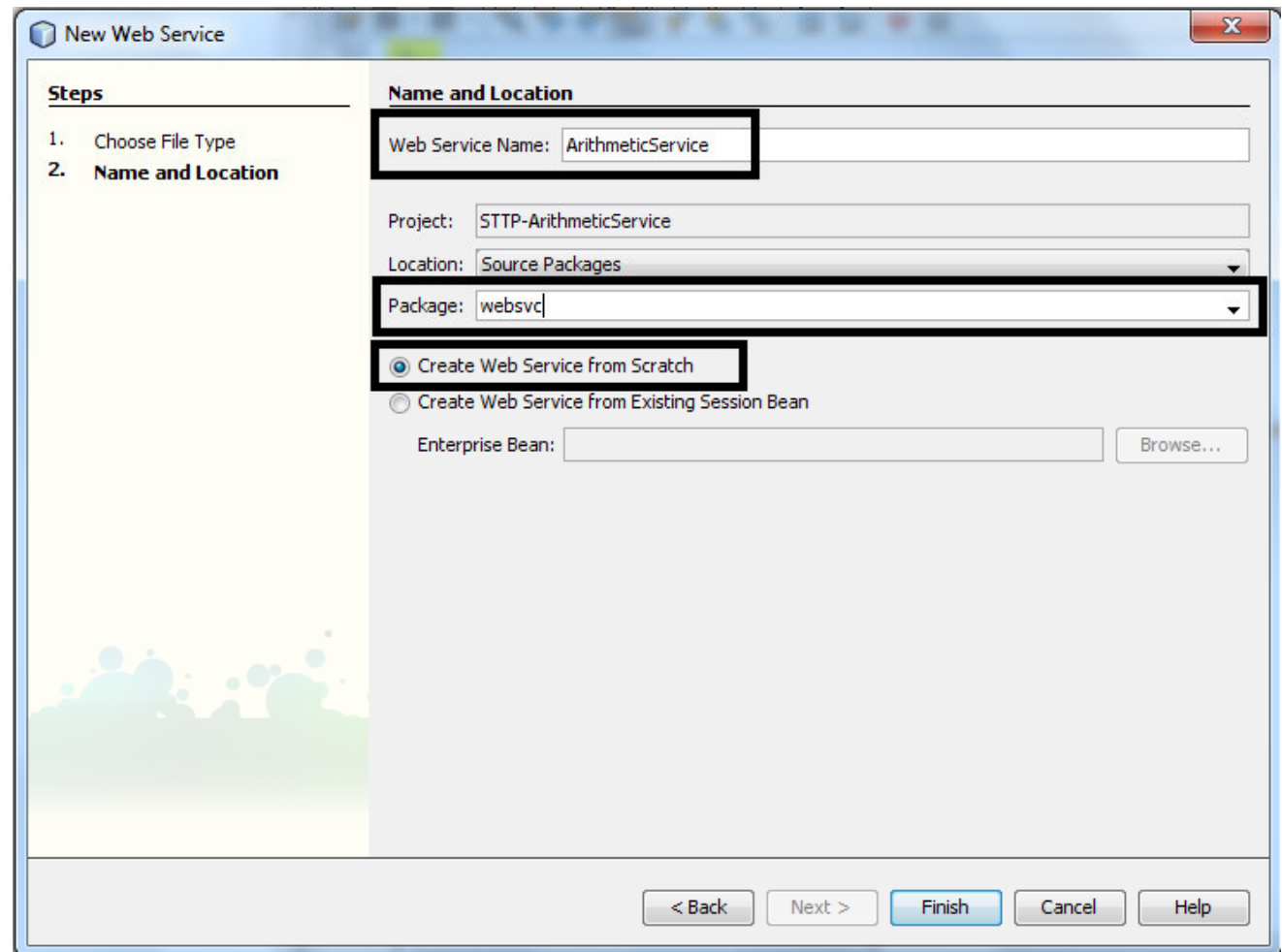
Creating Web Service using Code First Approach

- Right click on project folder and select New-> Web Service



Creating Web Service using Code First Approach

- Provide Web Service Name, Package Name

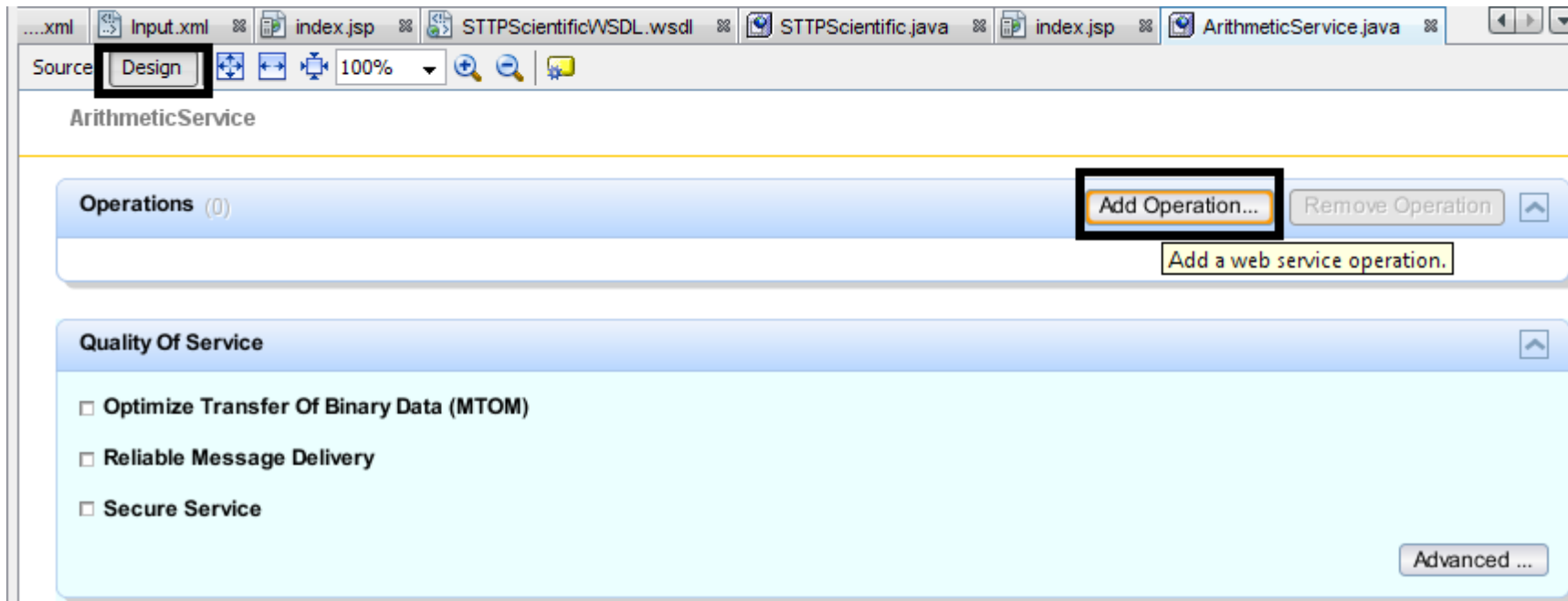


The screenshot shows the 'New Web Service' dialog box with the following details:

- Steps:**
 1. Choose File Type
 2. **Name and Location**
- Name and Location:**
 - Web Service Name: ArithmeticService
 - Project: STTP-ArithmeticService
 - Location: Source Packages
 - Package: websvc
 - ☒ Create Web Service from Scratch
 - ☐ Create Web Service from Existing Session Bean
 - Enterprise Bean: [Empty field] [Browse...]
- Buttons:** < Back, Next >, Finish, Cancel, Help

Creating Web Service using Code First Approach

- Go to Design View -> Click Add Operation



Creating Web Service using Code First Approach

Provide Following Details: Operation Name, Parameters and Return Type

Add Operation...

Name:

Return Type:

Parameters Exceptions

Name	Type	Final
Operand1	double	<input type="checkbox"/>
Operand2	java.lang.String	<input type="checkbox"/>

boolean
int
char
byte
short
long
float
double

Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.

Creating Web Service using Code First Approach

Note that the operation is added under the Web Service

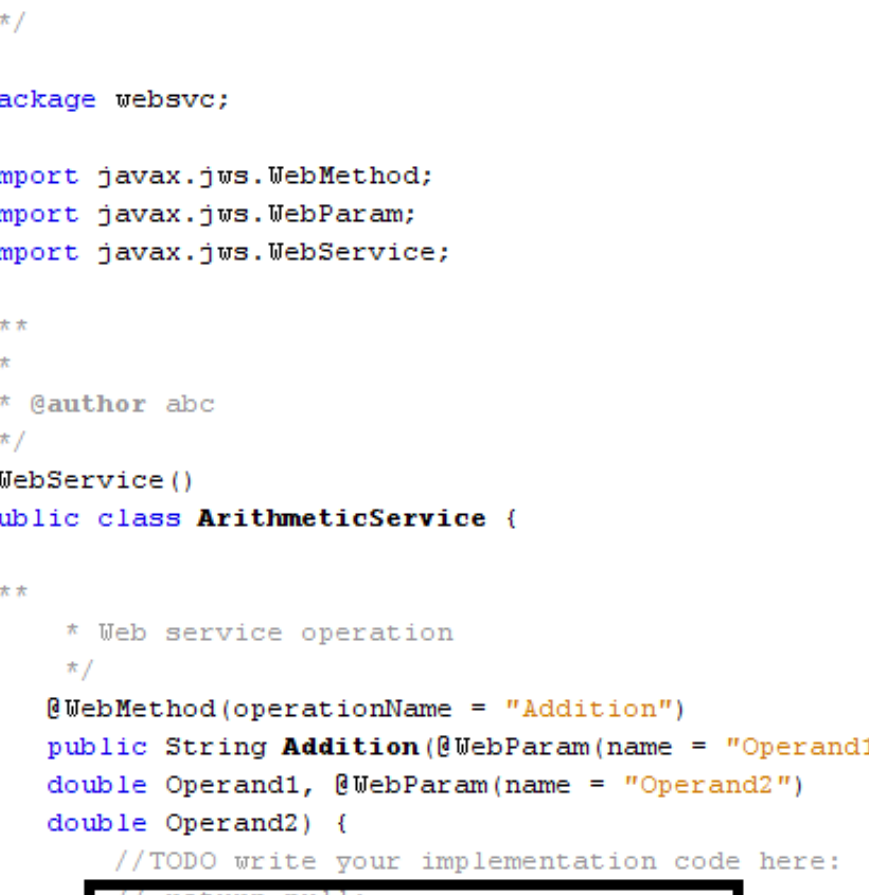
The screenshot shows the Eclipse IDE with the 'ArithmeticService' web service configuration. The 'Operations' tab is active, showing an 'Addition' operation. The parameters are listed in a table:

Parameter Name	Parameter Type
Operand1	double
Operand2	double

Below the parameters table, the 'Quality Of Service' section is visible, showing three checkboxes: 'Optimize Transfer Of Binary Data (MTOM)', 'Reliable Message Delivery', and 'Secure Service'. An 'Advanced ...' button is located at the bottom right of this section.

Creating Web Service using Code First Approach

Go to the Source View of Web Service and Modify the Code of Addition Operation



```
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/

package websvc;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;

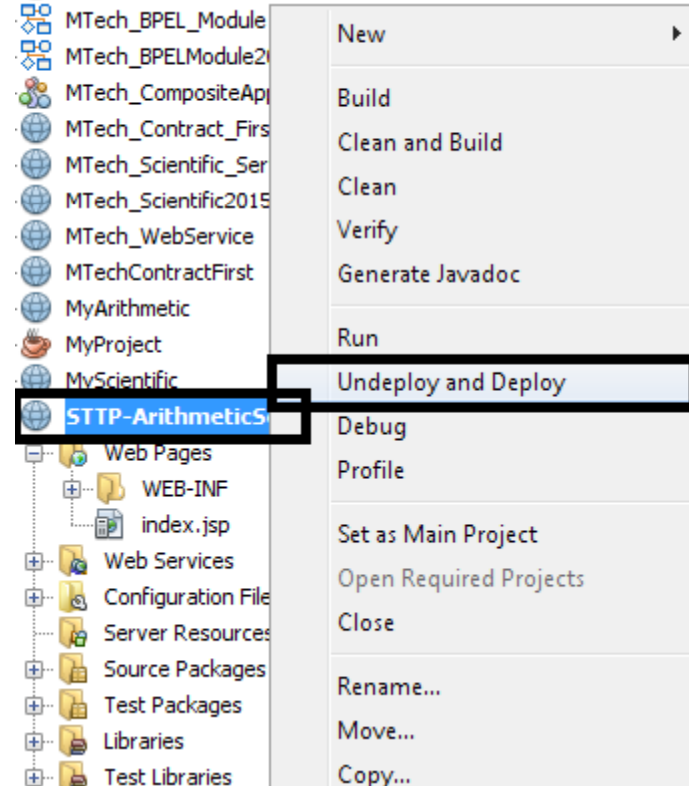
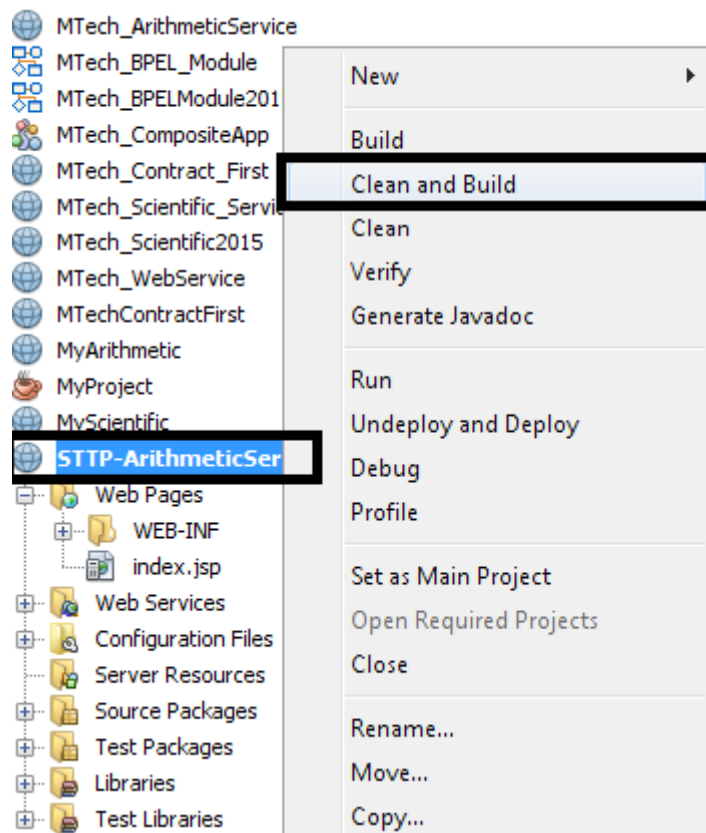
/**
 *
 * @author abc
 */
@WebService()
public class ArithmeticService {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "Addition")
    public String Addition(@WebParam(name = "Operand1")
        double Operand1, @WebParam(name = "Operand2")
        double Operand2) {
        //TODO write your implementation code here:
        // return null;
        double sum = Operand1 + Operand2;
        return sum+"";
    }
}
```

Creating Web Service using Code First Approach

Clean and Build the Project

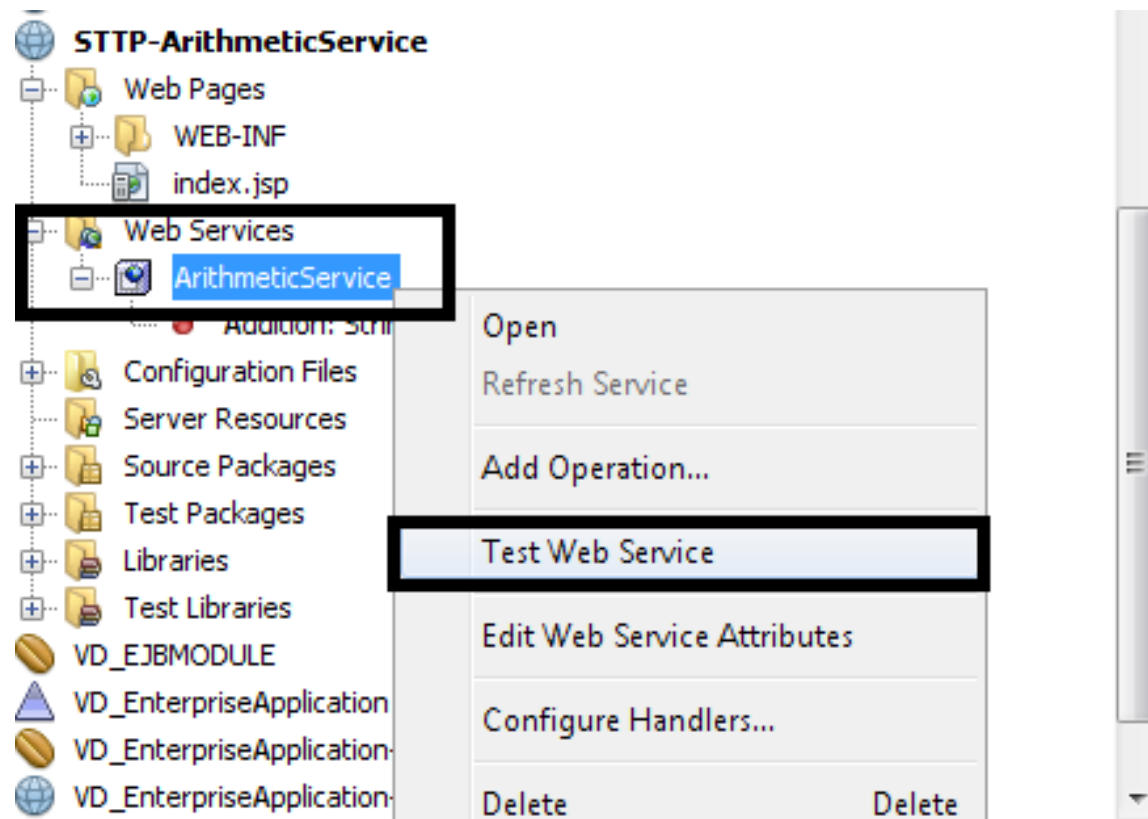
Deploy the Project



Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.

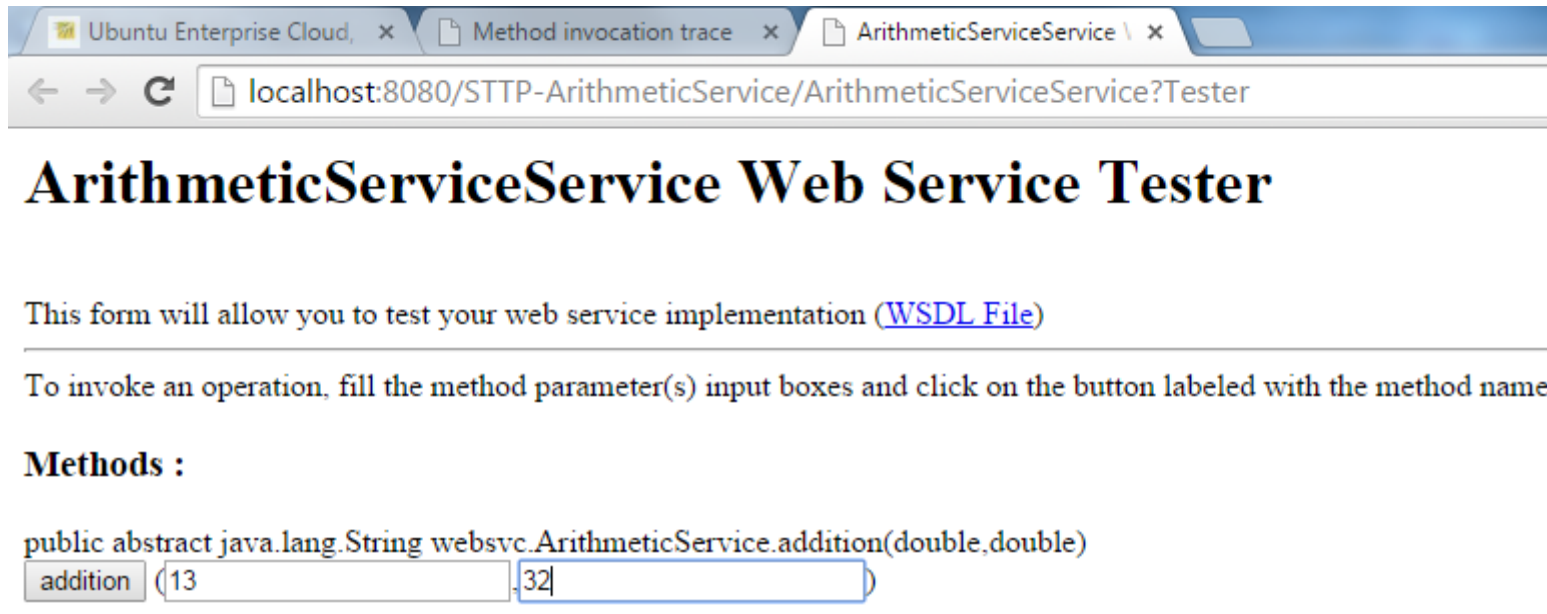
Creating Web Service using Code First Approach

Select Project -> Web Services -> ArithmeticService -> Test Web Service



Creating Web Service using Code First Approach

Test Web Service



The screenshot shows a web browser window with three tabs: 'Ubuntu Enterprise Cloud', 'Method invocation trace', and 'ArithmeticServiceService'. The address bar displays 'localhost:8080/STTP-ArithmeticService/ArithmeticServiceService?Tester'. The main heading is 'ArithmeticServiceService Web Service Tester'. Below the heading, a text block states: 'This form will allow you to test your web service implementation ([WSDL File](#))'. A horizontal line follows. Another text block says: 'To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.' Below this, the section 'Methods :' is shown. Under 'Methods :', the Java code snippet 'public abstract java.lang.String webservice.ArithmeticService.addition(double,double)' is displayed. Below the code, there is a button labeled 'addition' and two input fields. The first input field contains the value '13' and the second input field contains the value '32'.

ArithmeticServiceService Web Service Tester

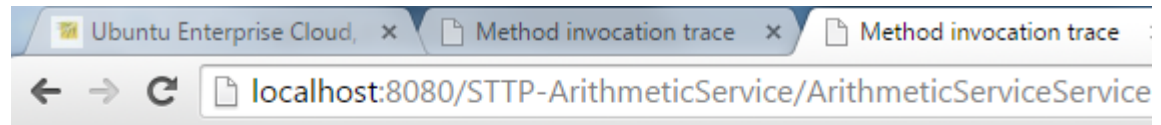
This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String webservice.ArithmeticService.addition(double,double)

addition (13 32)



addition Method invocation

Method parameter(s)

Type	Value
double	13
double	32

Method returned

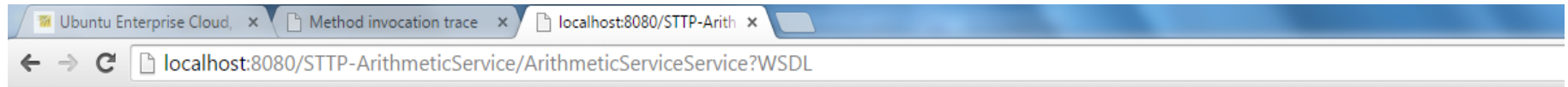
java.lang.String : "45.0"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:Addition xmlns:ns2="http://websvc/">
      <Operand1>13.0</Operand1>
      <Operand2>32.0</Operand2>
    </ns2:Addition>
  </S:Body>
</S:Envelope>
```

SOAP Response

Look at WSDL File



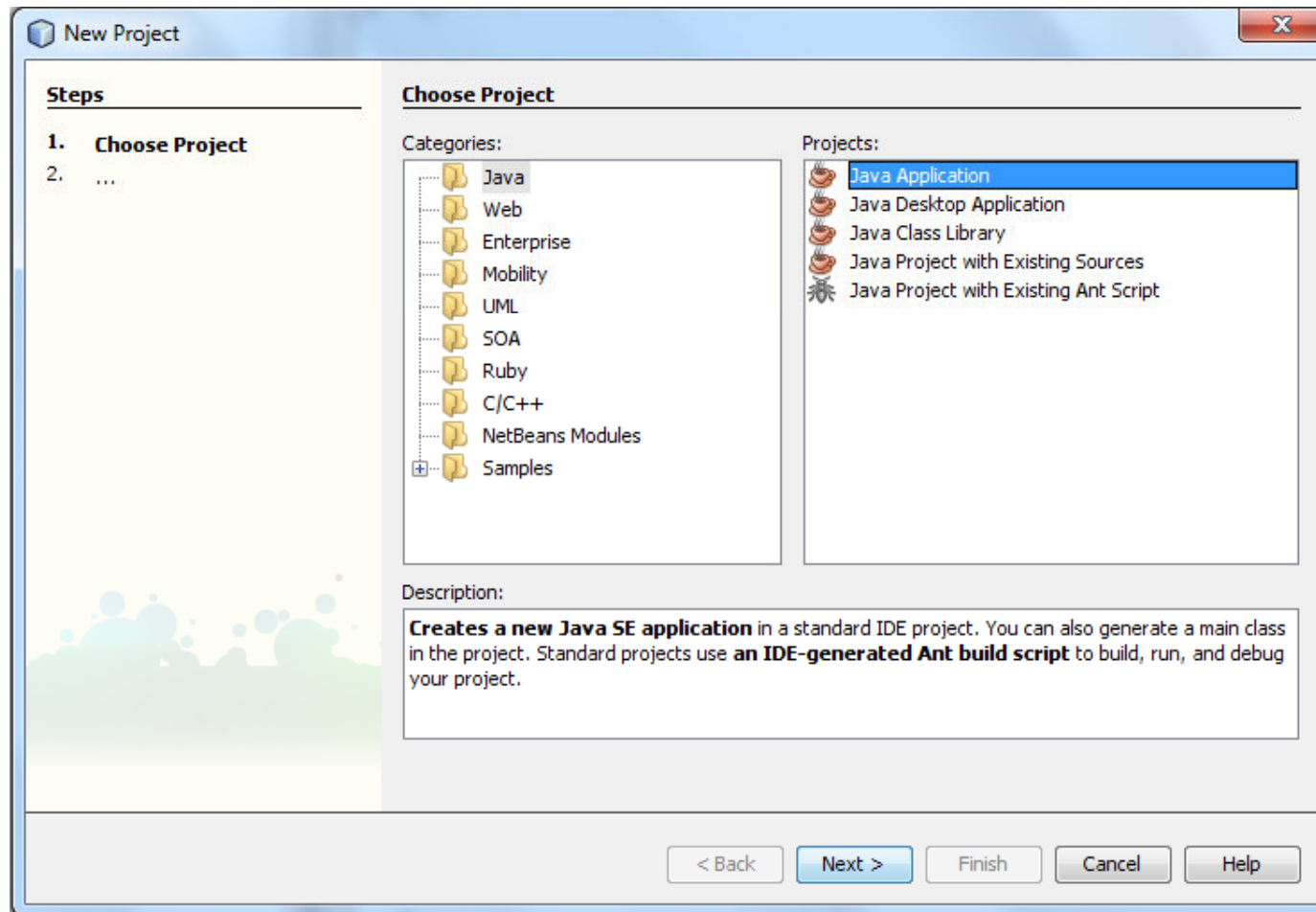
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.2-hudson-182-RC1.
-->
<!--
  Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.2-hudson-182-RC1.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://websvc/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://websvc/" name="ArithmeticServiceService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://websvc/" schemaLocation="http://localhost:8080/STTP-ArithmeticService/ArithmeticServiceService?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="Addition">
    <part name="parameters" element="tns:Addition"/>
  </message>
  <message name="AdditionResponse">
    <part name="parameters" element="tns:AdditionResponse"/>
  </message>
  <portType name="ArithmeticService">
    <operation name="Addition">
      <input message="tns:Addition"/>
      <output message="tns:AdditionResponse"/>
    </operation>
  </portType>
  <binding name="ArithmeticServicePortBinding" type="tns:ArithmeticService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="Addition">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
```

Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.

Building Web-Service Client

Create a New Java Project



Building Web-Service Client

Give Project Name – STTP-Client

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

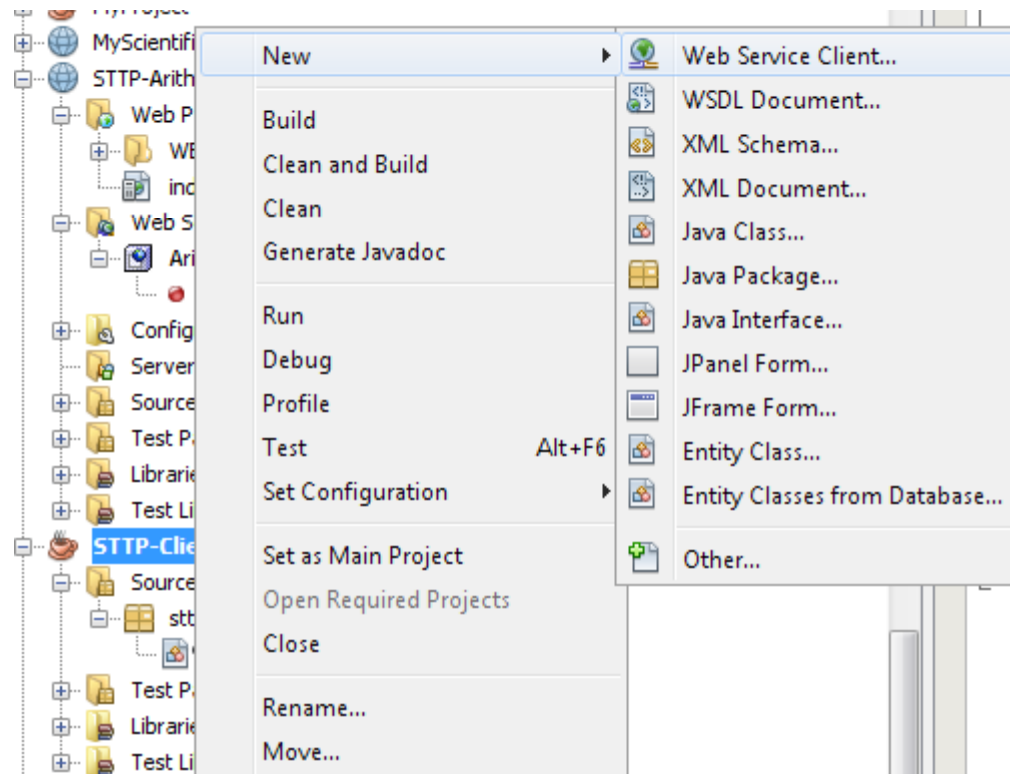
☒ Set as Main Project

☒ Create Main Class

< Back Next > **Finish** Cancel Help

Building Web-Service Client

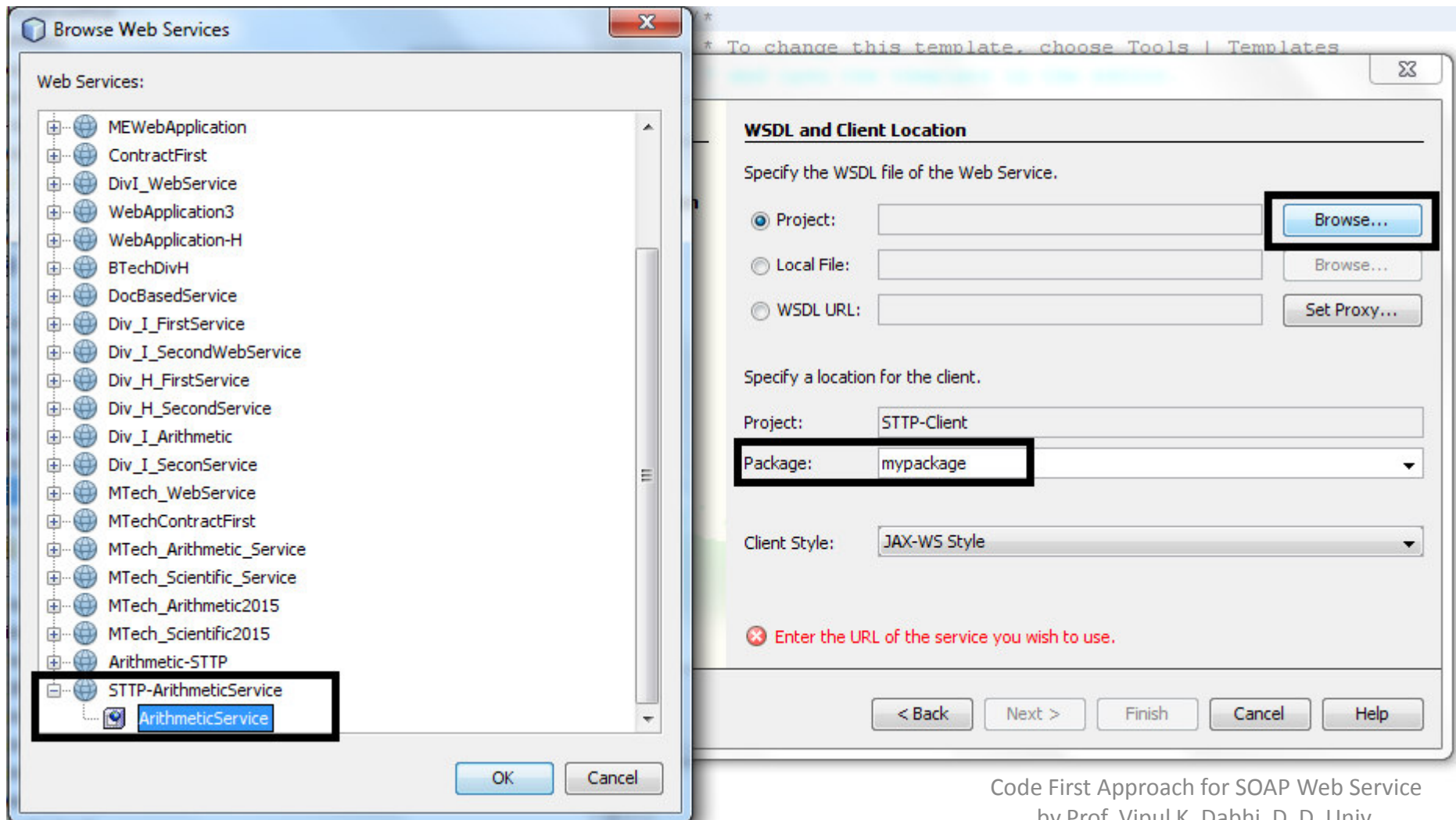
Select Project -> New -> Web Service Client



Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.

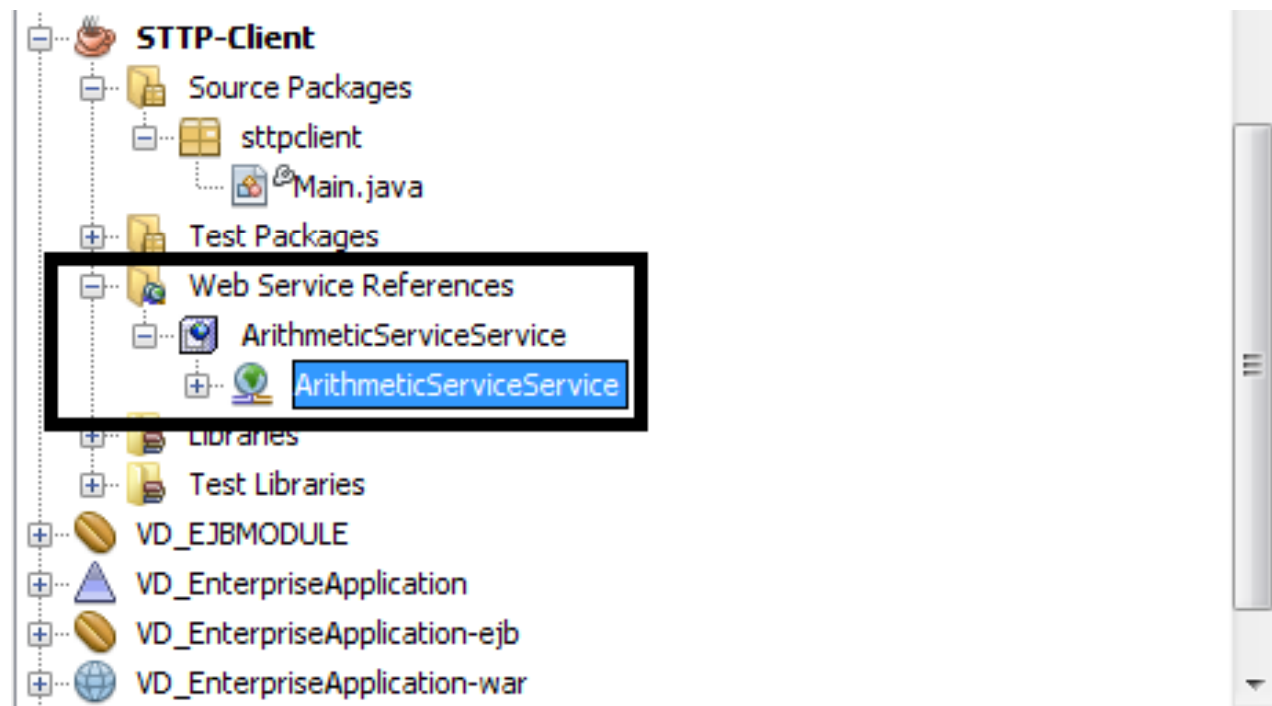
Building Web-Service Client

Specify WSDL Location and Package Name



Building Web-Service Client

The Web Service Reference gets Added into the project



Building Web-Service Client

Go to source code of Main.java – Right click -> Web Service Client Resources -> Call Web Service Operation

The screenshot shows an IDE interface. On the left, the Project Explorer displays a project named 'STTP-Client' with a sub-package 'sttpclient' containing 'Main.java', which is highlighted with a black box. The main editor window shows the source code of 'Main.java'. The code includes a package declaration 'package sttpclient;', a class comment '/* To change this template, choose Tools | Templates * and open the template in the editor. */', and a class definition 'public class Main {'. Inside the class, there is a comment '/* @param args the command line arguments */' and a method signature 'public static void main(String[] args) {' with a placeholder comment '// TODO code application logic here'. A right-click context menu is open over the code, showing various actions. At the bottom of the menu, 'Call Web Service Operation...' and 'Web Service Client Resources' are highlighted with black boxes. The status bar at the bottom shows '19:48' and 'INS'.

```
/* To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package sttpclient;

/**
 *
 * @author abc
 */
public class Main {

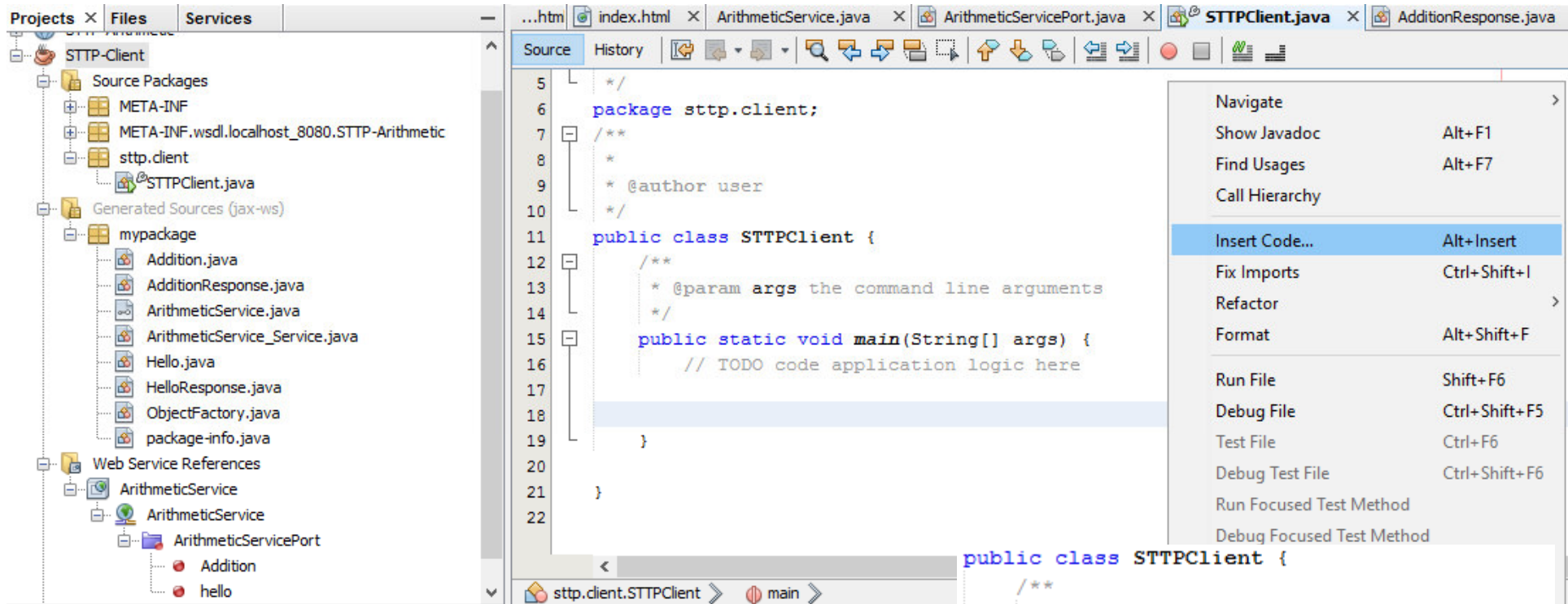
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Call Web Service Operation... Web Service Client Resources

Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.

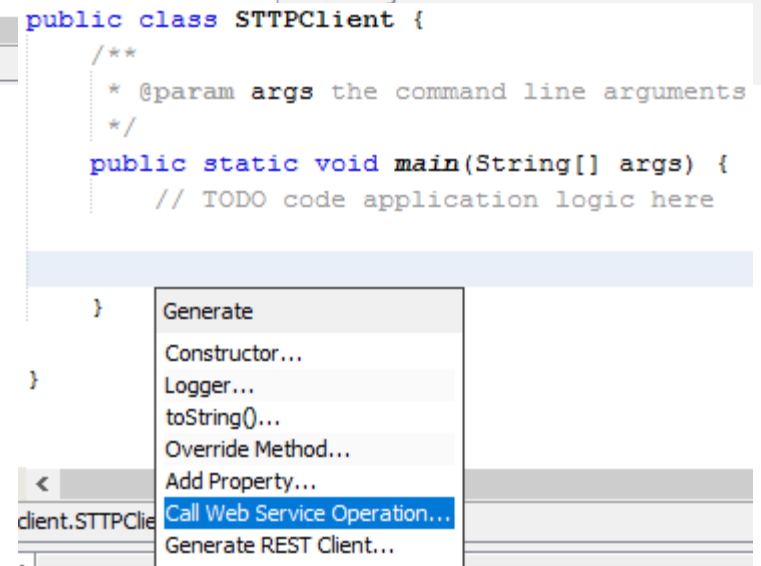
Building Web-Service Client



Netbeans 8.2:

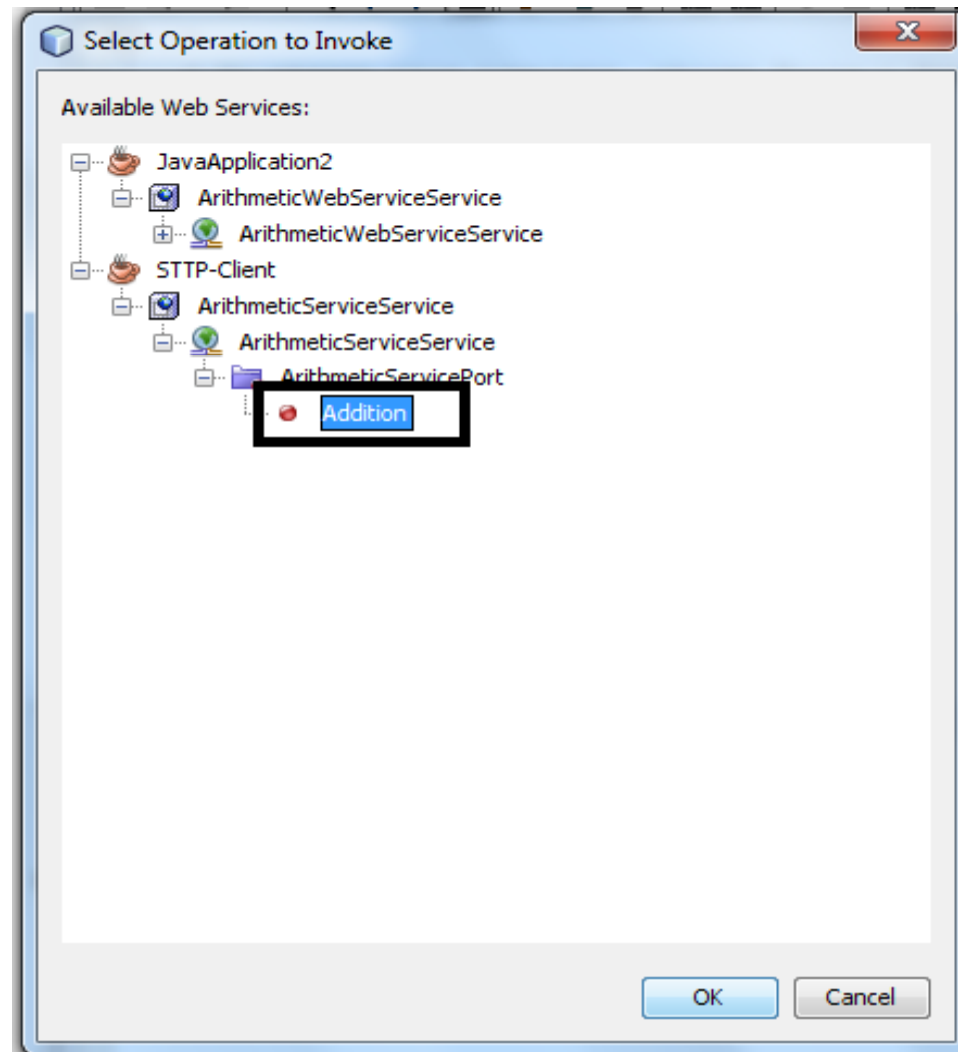
Go to source code of STTPClient.java –
Right click -> Insert Code -> Call Web
Service Operation

Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.



Building Web-Service Client

Select STTP-Client Project. Select Operation Addition.



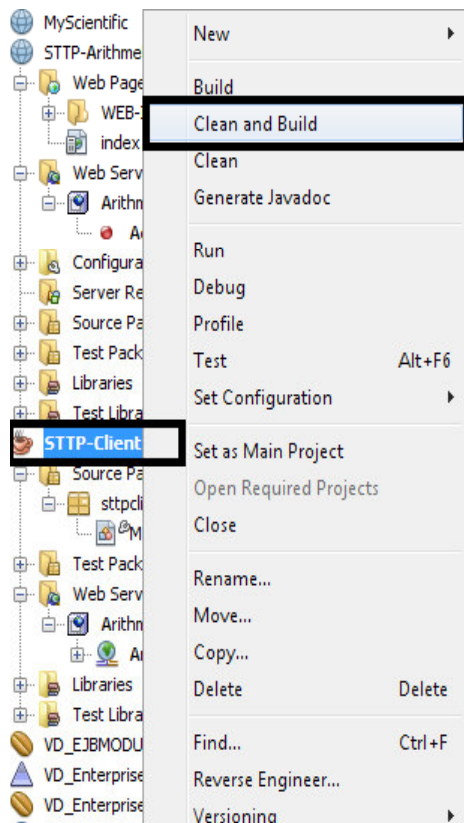
Building Web-Service Client

Notice that the code for calling Addition operation gets automatically added into source code of Main.java

```
public class Main {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
  
        try { // Call Web Service Operation  
            mypackage.ArithmeticServiceService service = new mypackage.ArithmeticServiceService();  
            mypackage.ArithmeticService port = service.getArithmeticServicePort();  
            // TODO initialize WS operation arguments here  
            double operand1 = 0.0;  
            double operand2 = 0.0;  
            // TODO process result here  
            java.lang.String result = port.addition(operand1, operand2);  
            System.out.println("Result = "+result);  
        } catch (Exception ex) {  
            // TODO handle custom exceptions here  
        }  
    }  
}
```


Building Web-Service Client

Change the values of parameters in source code of Main.java. Clean and Build Application.



Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.

Building Web-Service Client

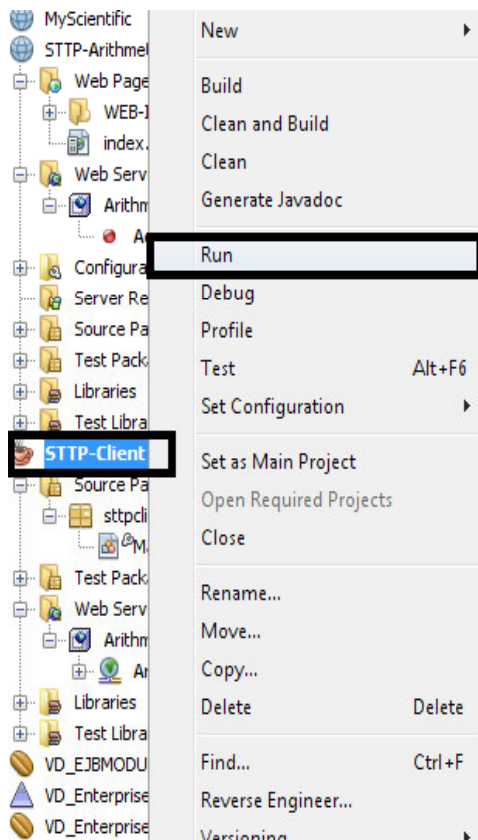
Netbeans 8.2

Notice that the code for calling addition operation gets automatically added into source code of STTPClient.java

```
public class STTPClient {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
  
        String result = addition (40,60);  
        System.out.println("Result="+result);  
    }  
  
    private static String addition(double operand1, double operand2) {  
        mypackage.ArithmeticService_Service service = new mypackage.ArithmeticService_Service();  
        mypackage.ArithmeticService port = service.getArithmeticServicePort();  
        return port.addition(operand1, operand2);  
    }  
}
```

Building Web-Service Client

Run the Application.



```
* @author abc
*/
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        try { // Call Web Service Operation
            mypackage.ArithmeticServiceService service = new mypackage.ArithmeticServiceService();
            mypackage.ArithmeticService port = service.getArithmeticServicePort();
            // TODO initialize WS operation arguments here
            double operand1 = 10.0;
            double operand2 = 20.0;
            // TODO process result here
            java.lang.String result = port.addition(operand1, operand2);
            System.out.println("Result = "+result);
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
    }
}
```

Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.

Building Web-Service Client

Observe the output in Console.



The screenshot displays the NetBeans IDE interface. On the left, the 'Project Explorer' shows the 'STTP-Client' project with sub-items like 'Source Packages' and 'sttpclient'. The main editor window shows a Java file with the following code:

```
// TODO code application logic here

try { // Call Web Service Operation
    mypackage.ArithmeticServiceService service = new mypackage.ArithmeticServiceService();
    mypackage.ArithmeticService port = service.getArithmeticServicePort();
    // TODO initialize WS operation arguments here
    double operand1 = 10.0;
    double operand2 = 20.0;
```

Below the editor, the 'Output' window is active, showing the following log messages:

```
wsimport-client-check-ArithmeticServiceService:
wsimport-client-ArithmeticServiceService:
wsimport-client-generate:
wsimport-client-compile:
Compiling 1 source file to C:\Users\abc\Documents\NetBeansProjects\STTP-Client\build\classes
run:
Result = 30.0
BUILD SUCCESSFUL (total time: 1 second)
```

Code First Approach for SOAP Web Service
by Prof. Vipul K. Dabhi, D. D. Univ.