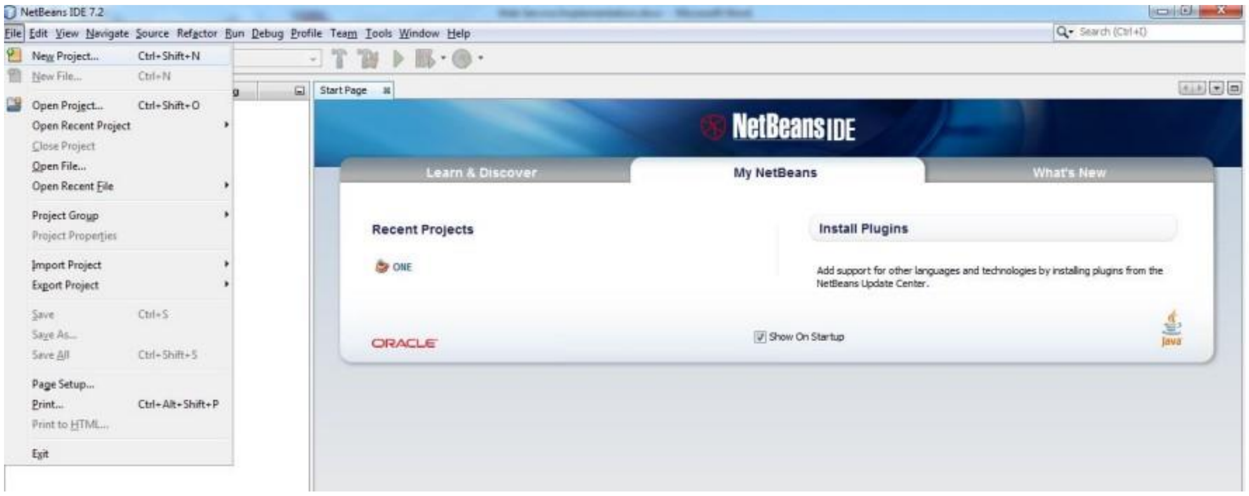
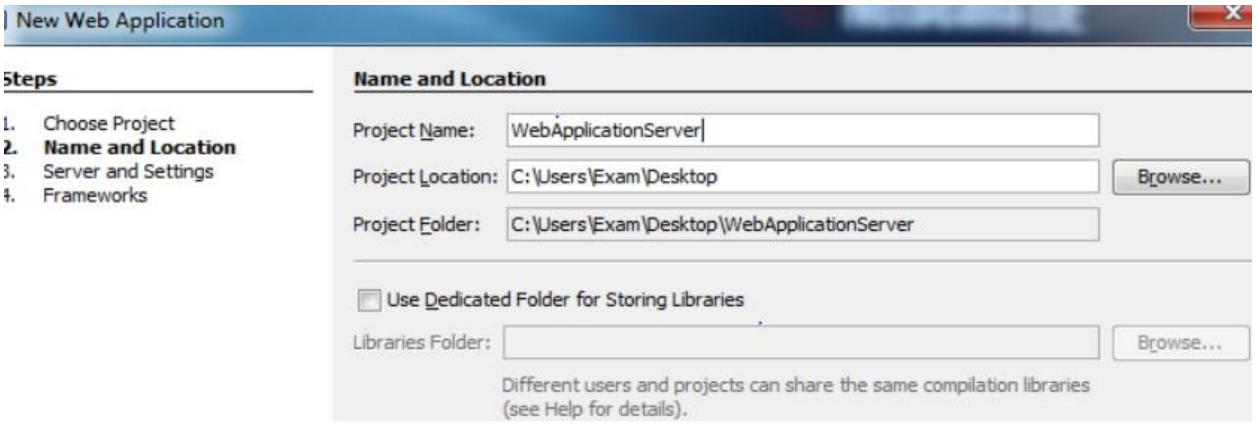



New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name: WebApplicationServer

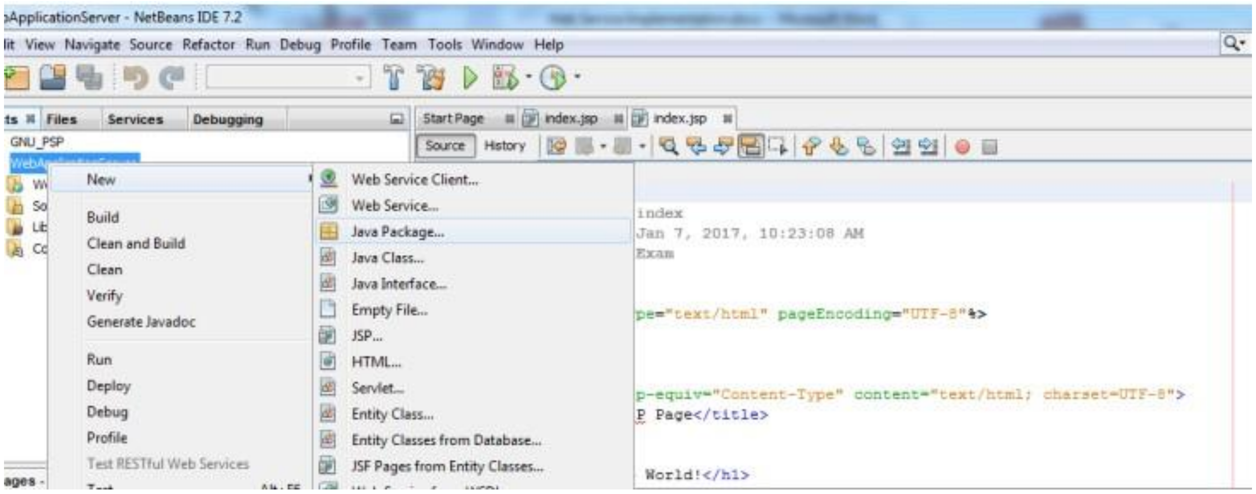
Project Location: C:\Users\Exam\Desktop

Project Folder: C:\Users\Exam\Desktop\WebApplicationServer

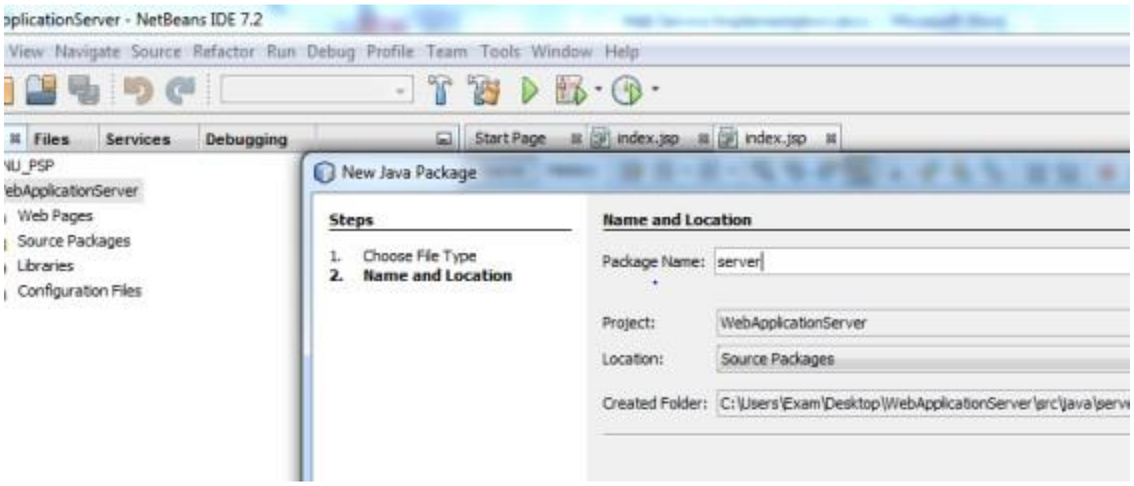
Use Dedicated Folder for Storing Libraries [ ]

Libraries Folder:

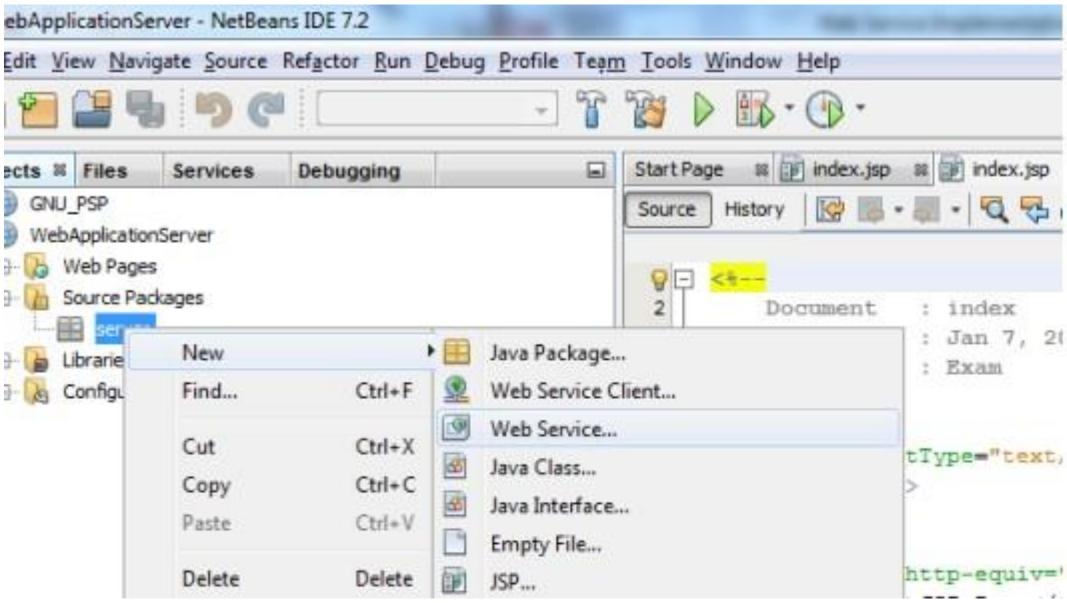
Different users and projects can share the same compilation libraries (see Help for details).

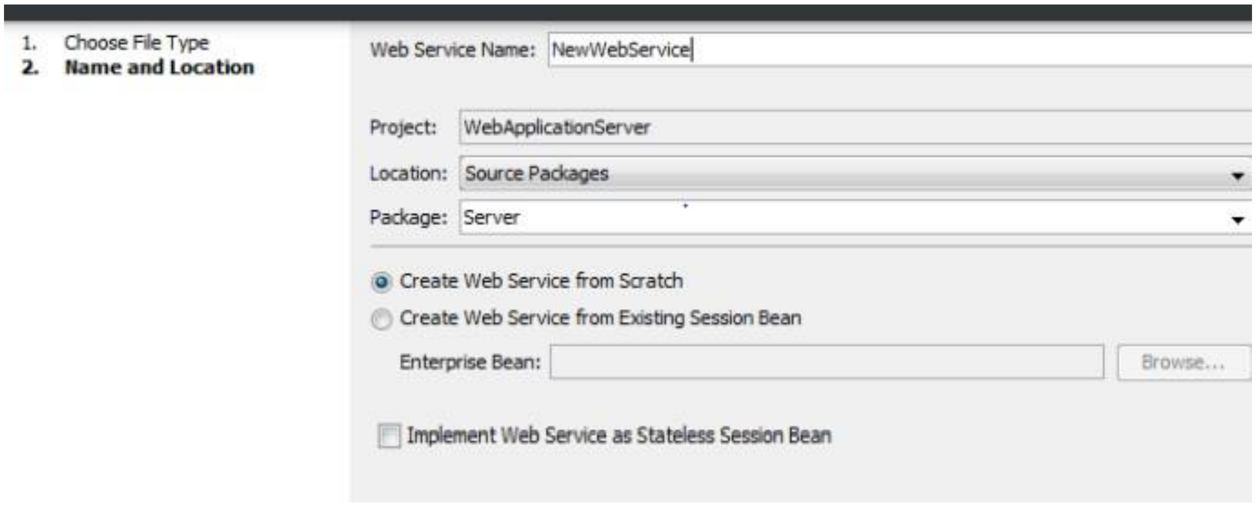


```
index
Jan 7, 2017, 10:23:08 AM
Exam

pe="text/html" pageEncoding="UTF-8"%>

p-equiv="Content-Type" content="text/html; charset=UTF-8">
P Page</title>

 World!</h1>
```



Package Name: server
Project: WebApplicationServer
Location: Source Packages
Created Folder: C:\Users\Exam\Desktop\WebApplicationServer\src\java\serve




1. Choose File Type
2. **Name and Location**

Web Service Name: NewWebService

Project: WebApplicationServer

Location: Source Packages

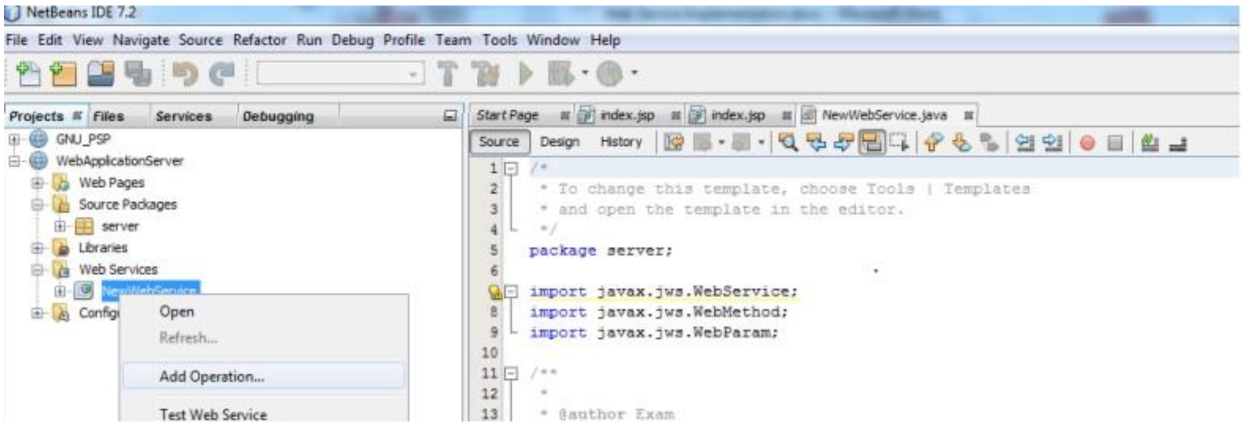
Package: Server

[✓ Create Web Service from Scratch]

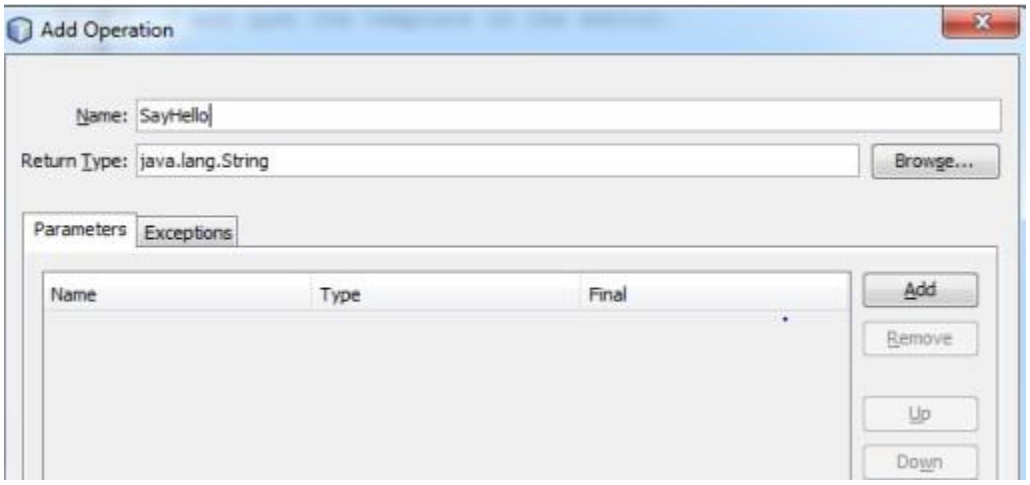
Create Web Service from Existing Session Bean

Enterprise Bean:

Implement Web Service as Stateless Session Bean



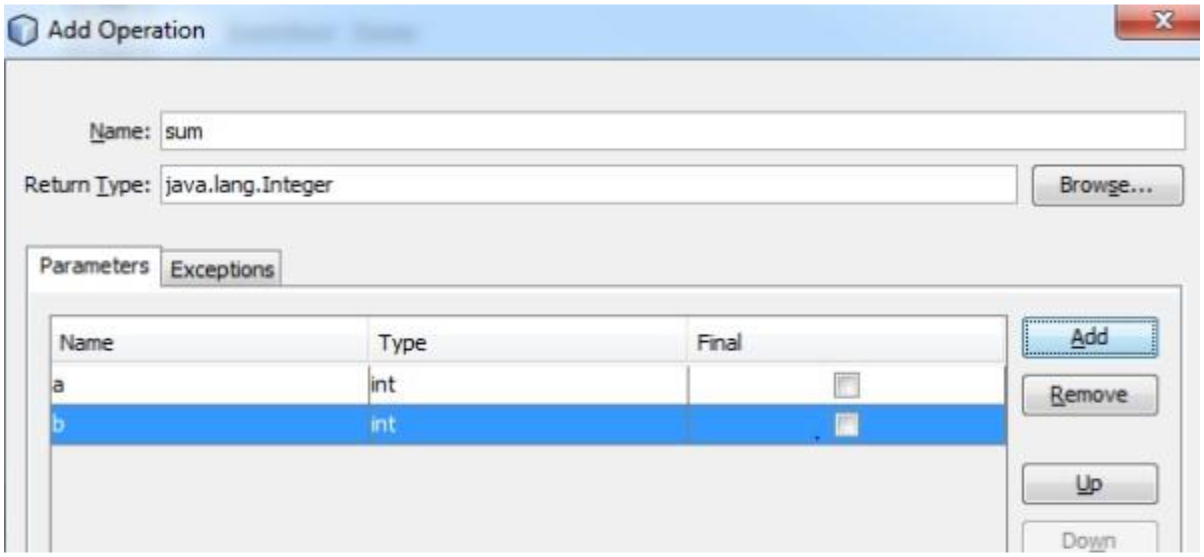
```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package server;

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

/**
 *
 * @author Exam
```



Name: SayHello

Return Type: java.lang.String

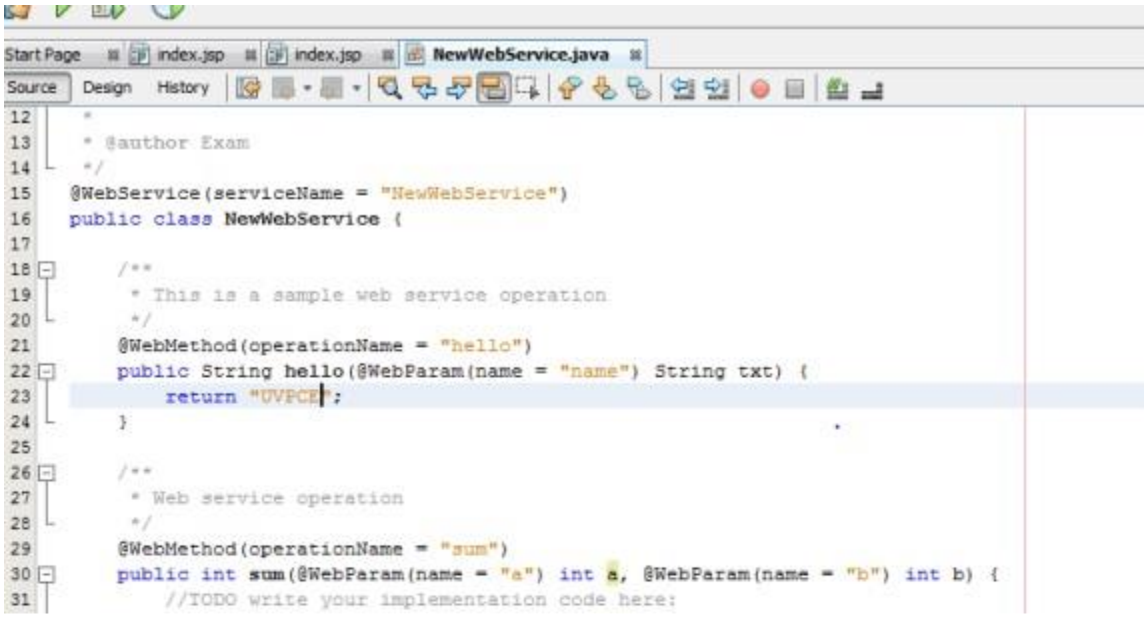
Name	Type	Final



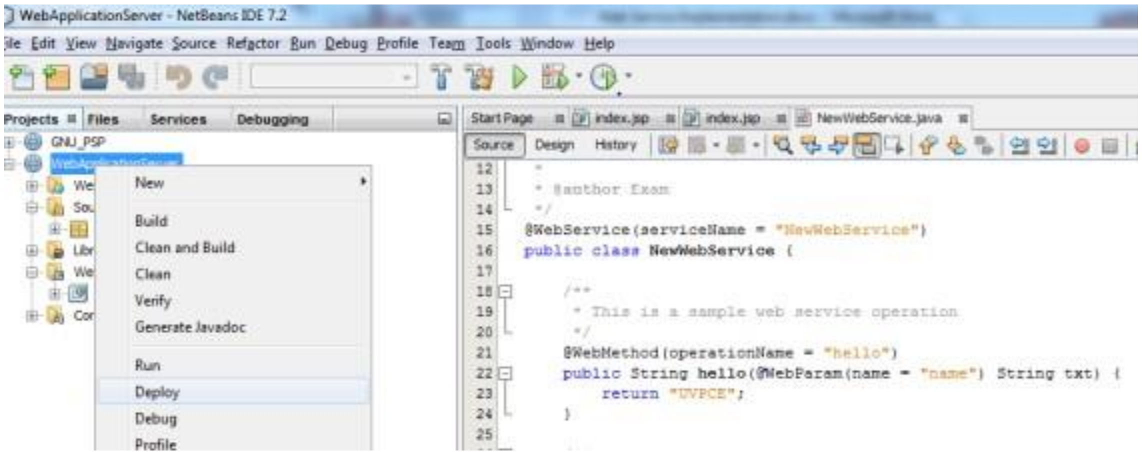
Name: sum

Return Type: java.lang.Integer

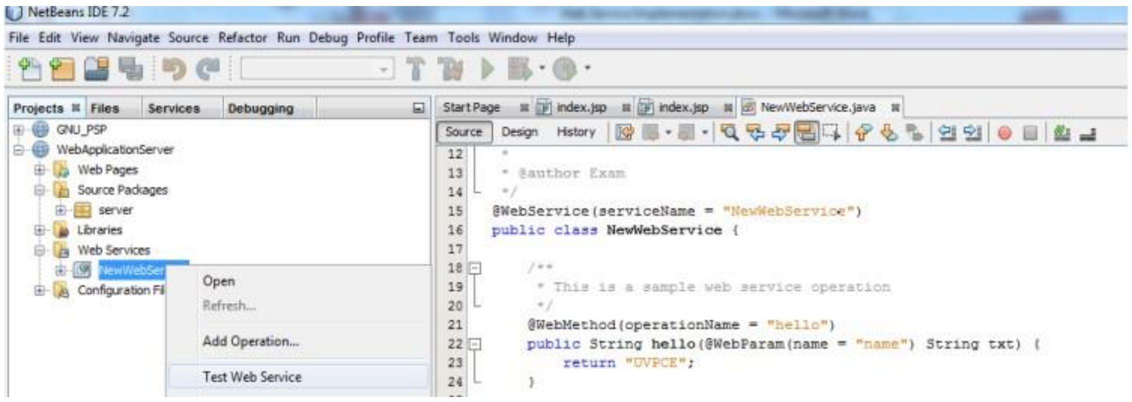
Name	Type	Final
a	int	☐
b	int	☐



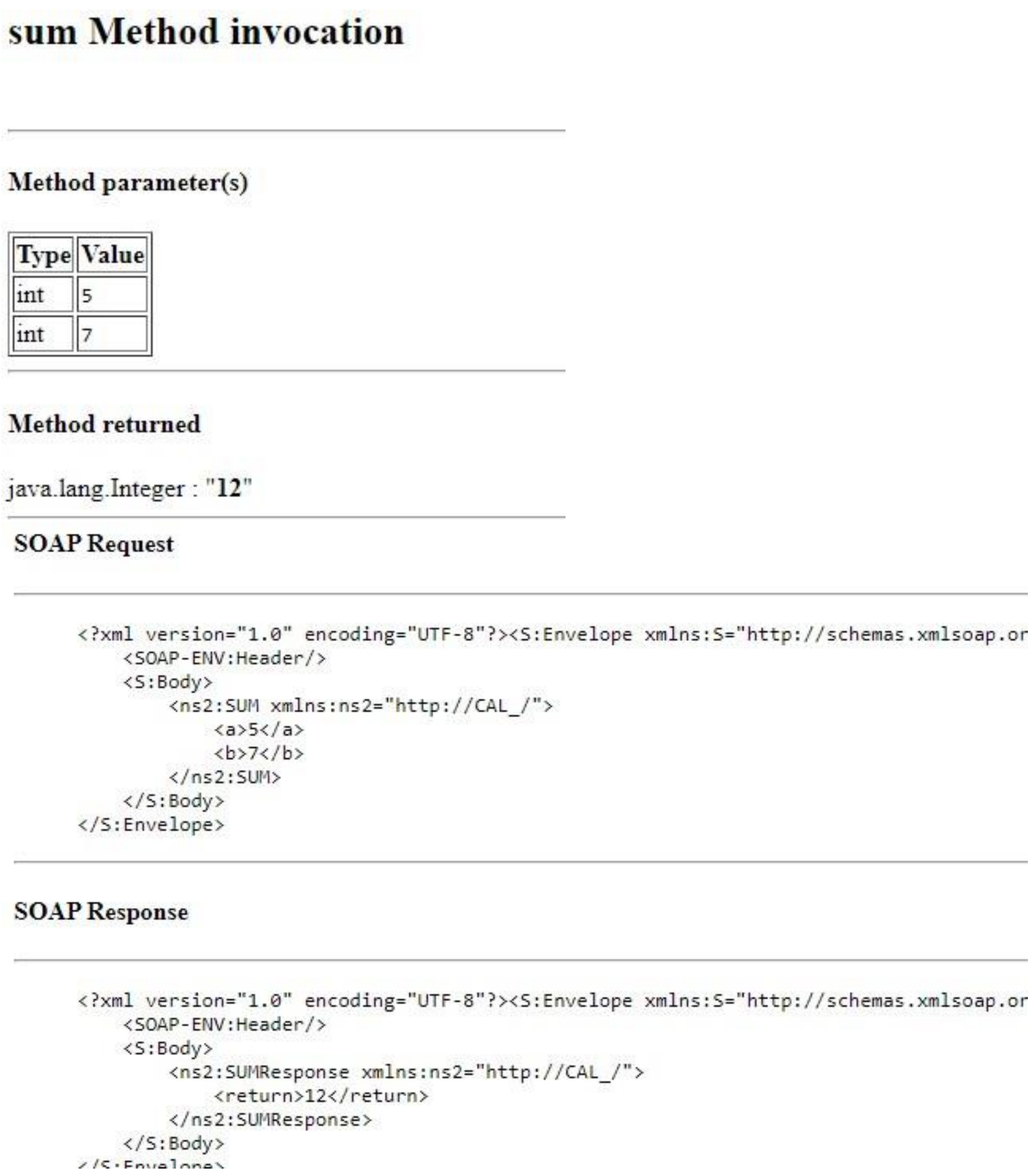
```
 *
 * @author Exam
 */
@WebService(serviceName = "NewWebService")
public class NewWebService {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "UVPCE";
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "sum")
    public int sum(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
        //TODO write your implementation code here:
```


WebApplicationServer - NetBeans IDE 7.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services Debugging
GNU_PSP
WebApplicationServer
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Start Page index.jsp index.jsp NewWebService.java
Source Design History

```
 *
 * @author Exam
 */
@WebService(serviceName = "NewWebService")
public class NewWebService {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "UVPCE";
    }
```



```
 *
 * @author Exam
 */
@WebService(serviceName = "NewWebService")
public class NewWebService {

    /**
     * This is a sample web service operation
     */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "UVPCE";
    }
```



sum Method invocation

Method parameter(s)

Type	Value
int	5
int	7

Method returned

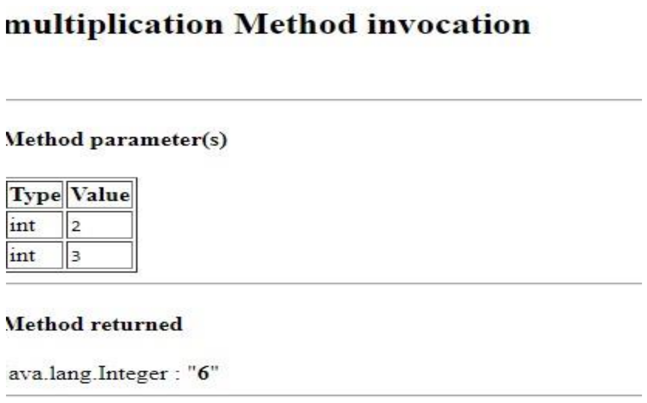
java.lang.Integer : "**12**"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.or
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:SUM xmlns:ns2="http://CAL_/">
            <a>5</a>
            <b>7</b>
        </ns2:SUM>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.or
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:SUMResponse xmlns:ns2="http://CAL_/">
            <return>12</return>
        </ns2:SUMResponse>
    </S:Body>
</S:Envelope>
```



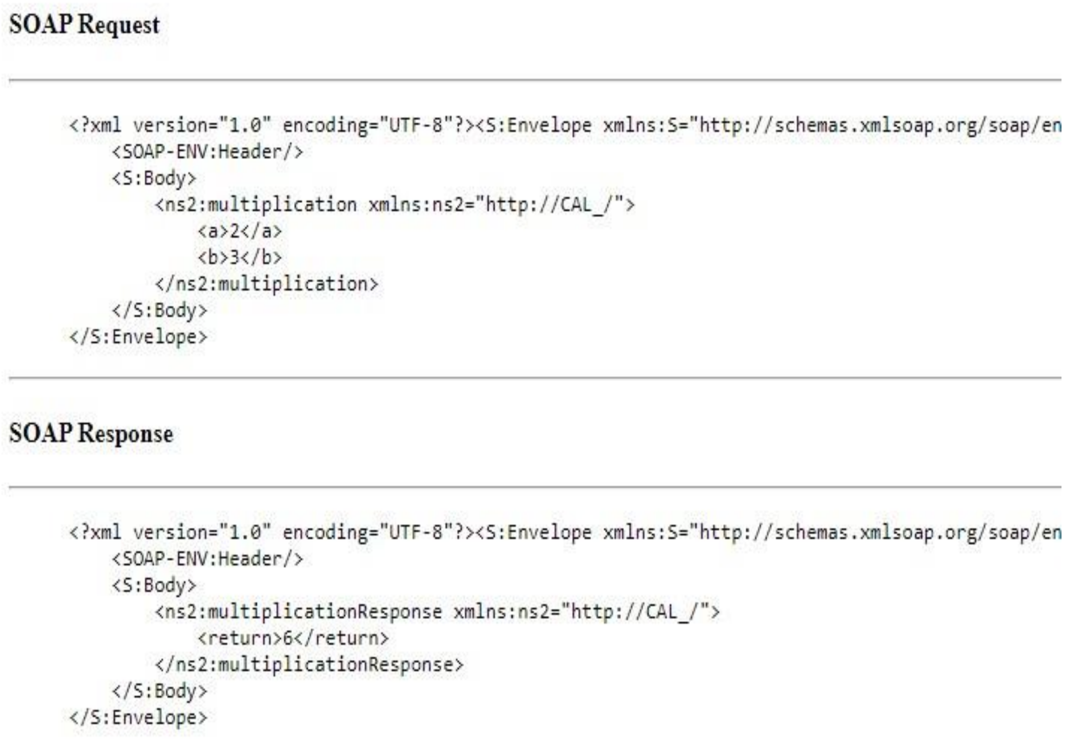
multiplication Method invocation

Method parameter(s)

Type	Value
int	2
int	3

Method returned

ava.lang.Integer : "**6**"

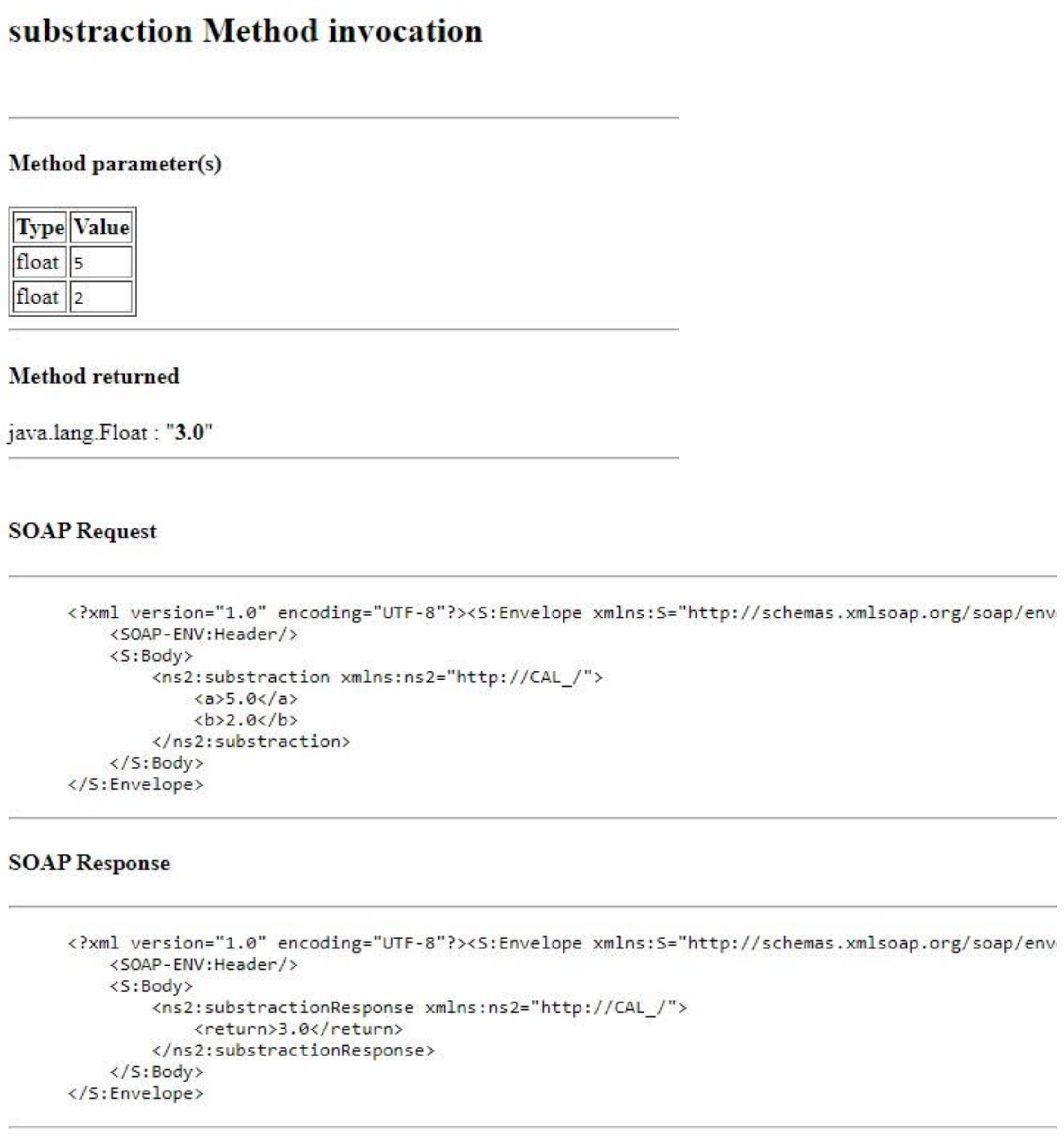


SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/en
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:multiplication xmlns:ns2="http://CAL_/">
            <a>2</a>
            <b>3</b>
        </ns2:multiplication>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/en
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:multiplicationResponse xmlns:ns2="http://CAL_/">
            <return>6</return>
        </ns2:multiplicationResponse>
    </S:Body>
</S:Envelope>
```



substraction Method invocation

Method parameter(s)

Type	Value
float	5
float	2

Method returned

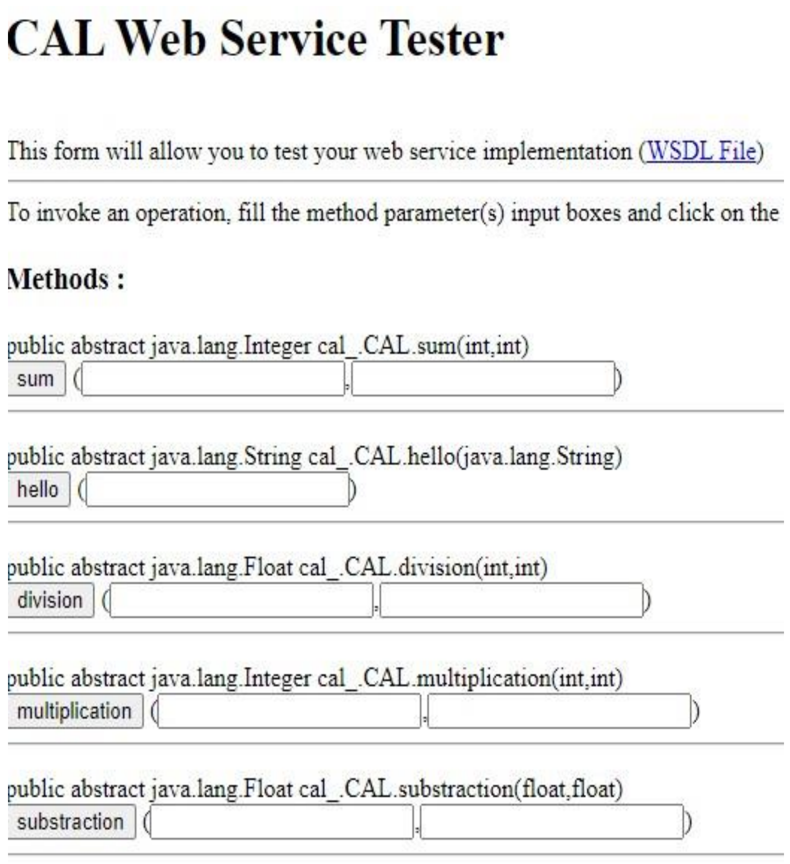
java.lang.Float : "**3.0**"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/env
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:substraction xmlns:ns2="http://CAL_/">
            <a>5.0</a>
            <b>2.0</b>
        </ns2:substraction>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/env
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:substractionResponse xmlns:ns2="http://CAL_/">
            <return>3.0</return>
        </ns2:substractionResponse>
    </S:Body>
</S:Envelope>
```



CAL 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

Methods :

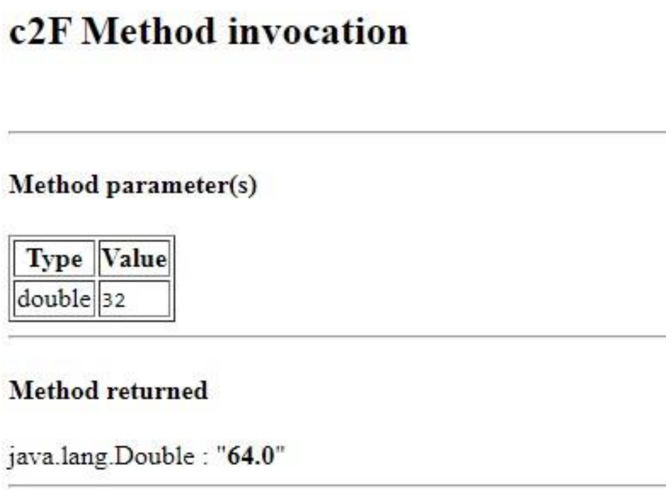
public abstract java.lang.Integer cal_.CAL.sum(int,int)
sum (,)

public abstract java.lang.String cal_.CAL.hello(java.lang.String)
hello ( )

public abstract java.lang.Float cal_.CAL.division(int,int)
division (,)

public abstract java.lang.Integer cal_.CAL.multiplication(int,int)
multiplication (,)

public abstract java.lang.Float cal_.CAL.substraction(float,float)
substraction (,)



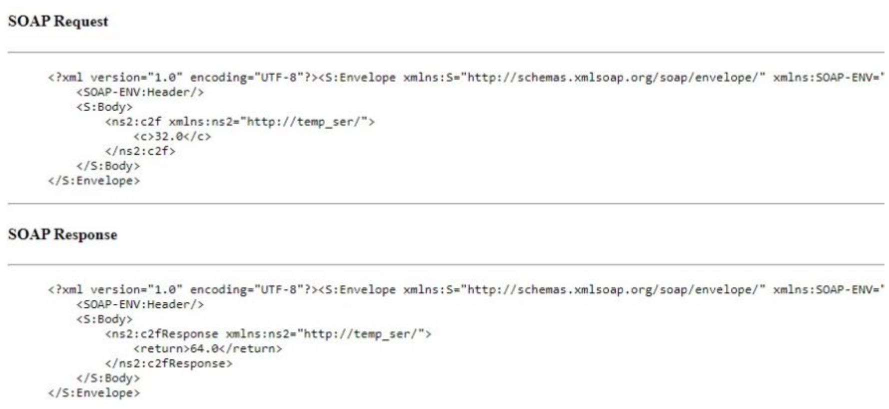
c2F Method invocation

Method parameter(s)

Type	Value
double	32

Method returned

java.lang.Double : "**64.0**"

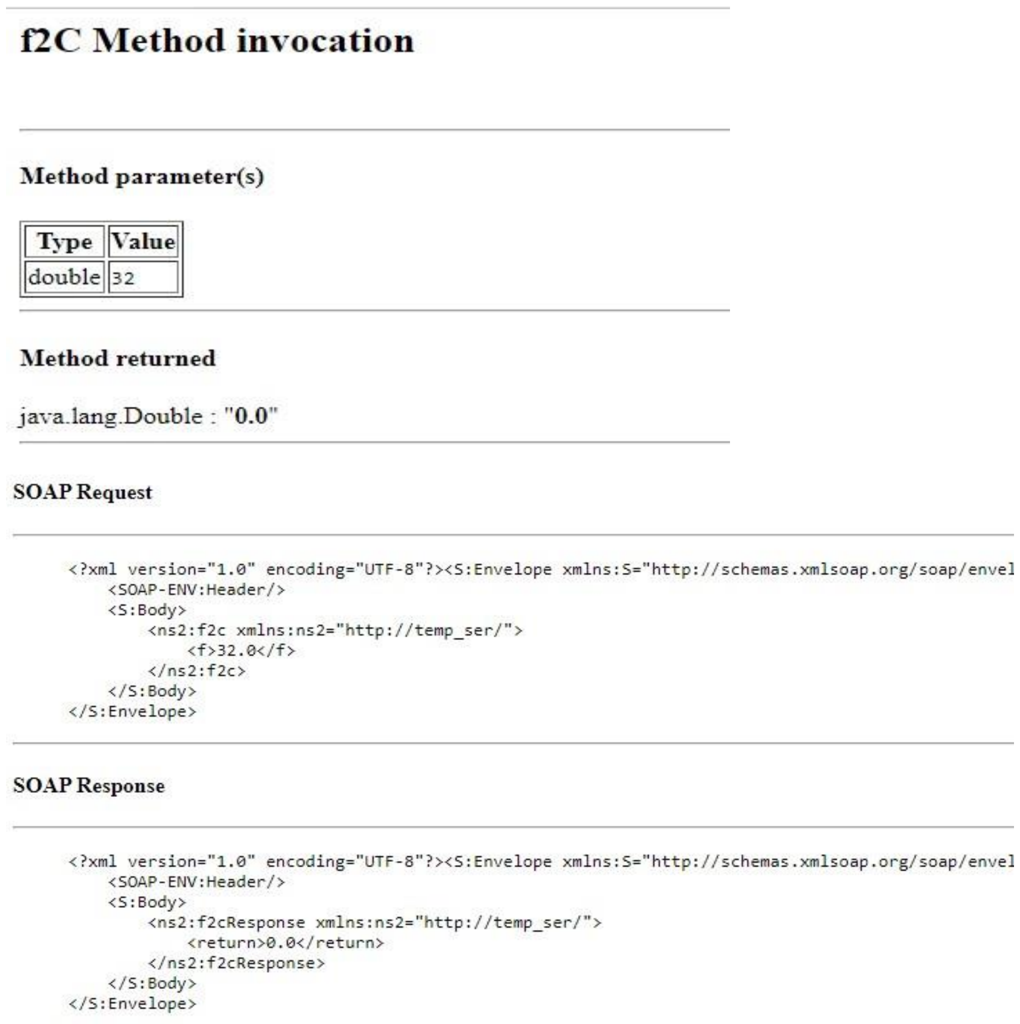


SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:c2f xmlns:ns2="http://temp_ser/">
            <c>32.0</c>
        </ns2:c2f>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:c2fResponse xmlns:ns2="http://temp_ser/">
            <return>64.0</return>
        </ns2:c2fResponse>
    </S:Body>
</S:Envelope>
```



f2C Method invocation

Method parameter(s)

Type	Value
double	32

Method returned

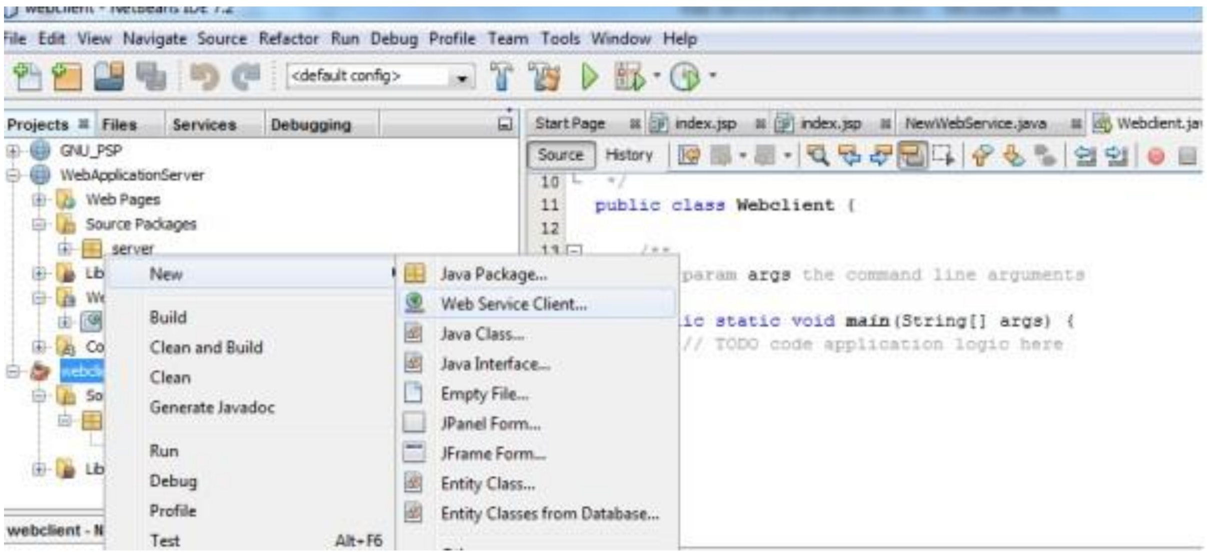
java.lang.Double : "**0.0**"

SOAP Request

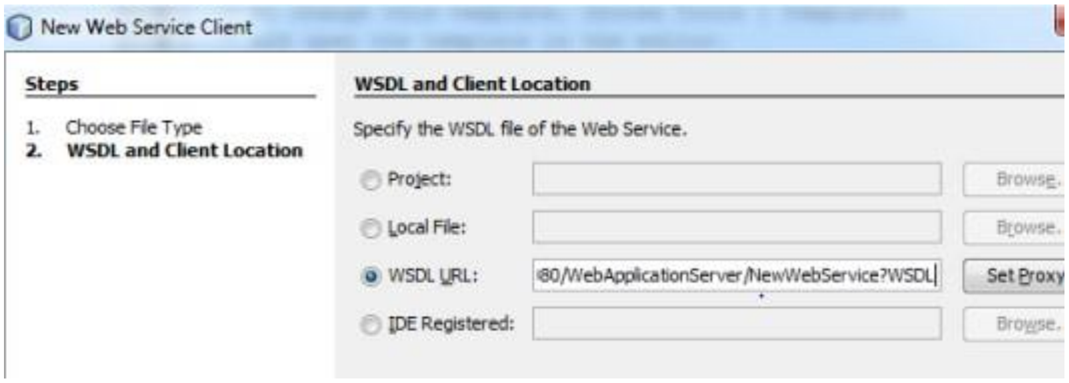
```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envel
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:f2c xmlns:ns2="http://temp_ser/">
            <f>32.0</f>
        </ns2:f2c>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envel
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:f2cResponse xmlns:ns2="http://temp_ser/">
            <return>0.0</return>
        </ns2:f2cResponse>
    </S:Body>
</S:Envelope>
```



```
*/
public class Webclient {

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



WSDL and Client Location

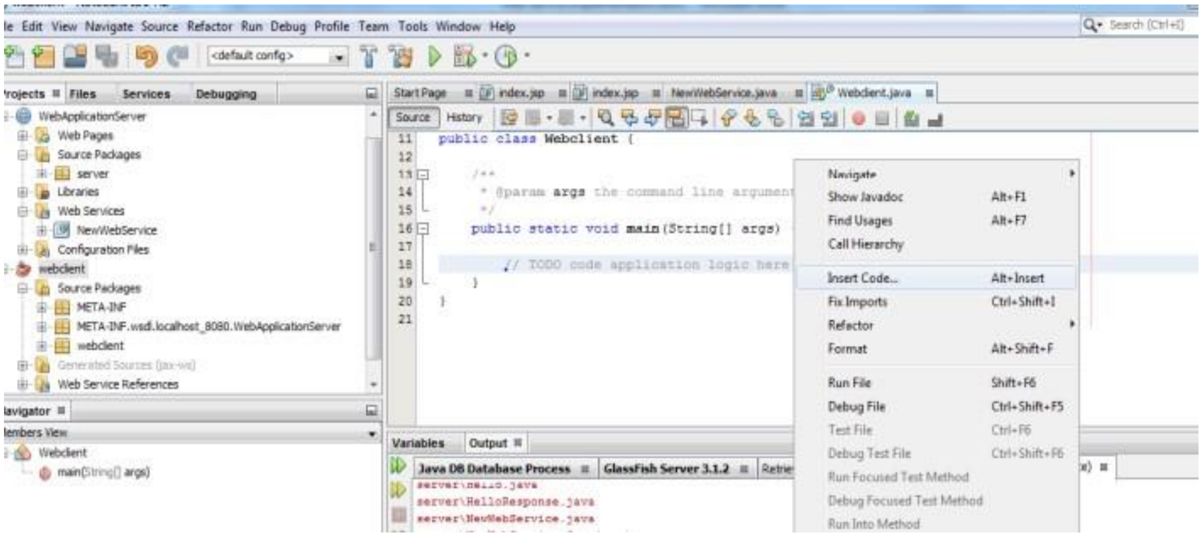
Specify the WSDL file of the Web Service.

Project:

Local File:

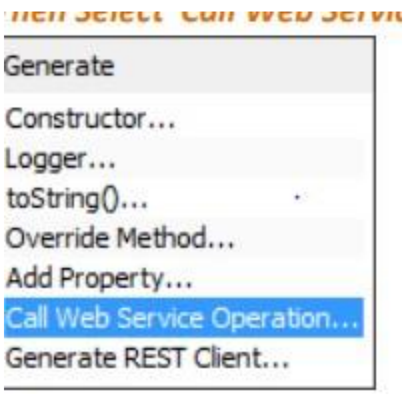
WSDL URL: 80/WebApplicationServer/NewWebService?WSDL

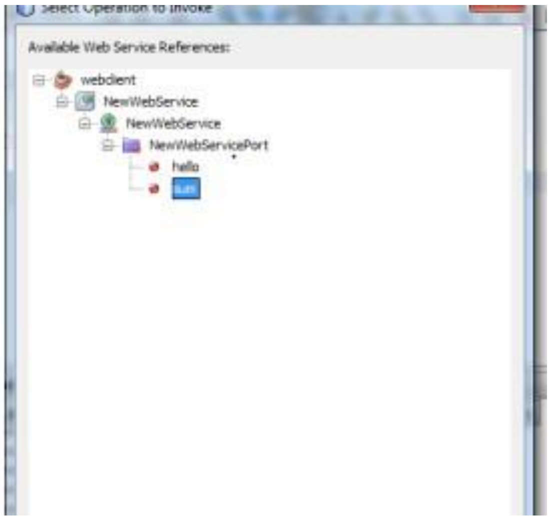
IDE Registered:

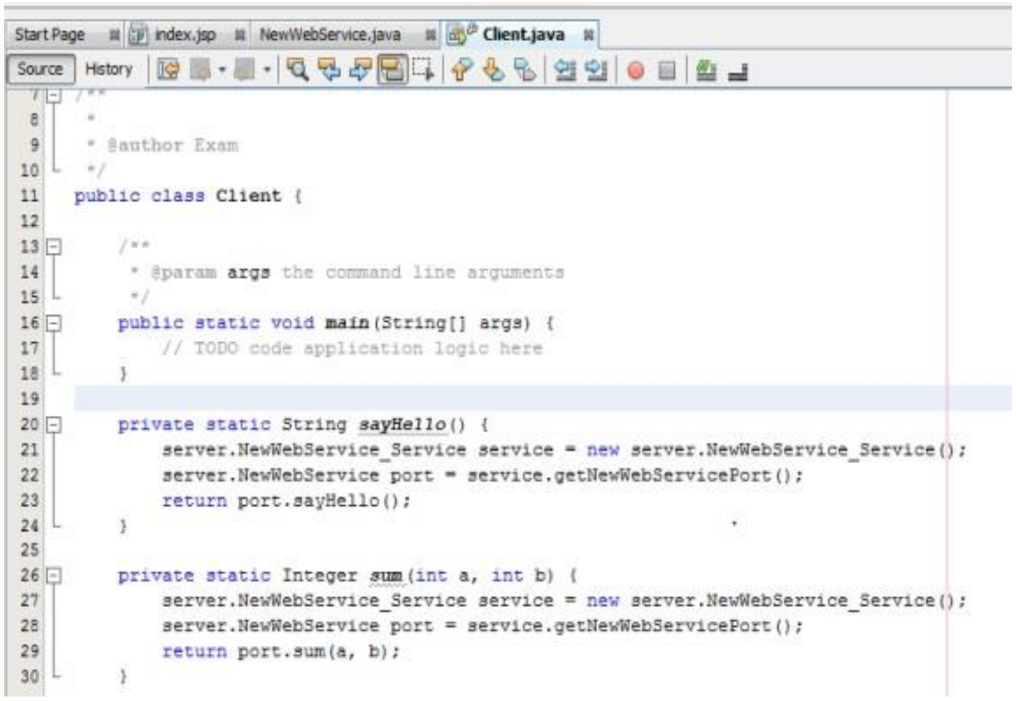


```
public class Webclient {

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

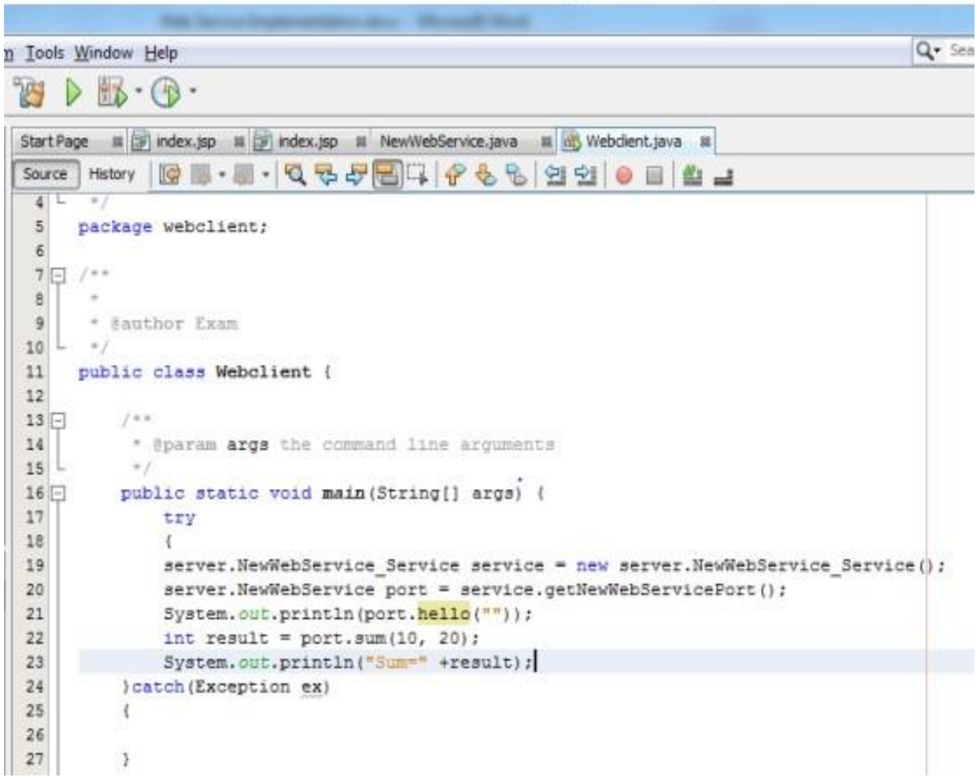
```
server\Hello.java
server\HelloResponse.java
server\NewWebService.java
```


Generate
Constructor...
Logger...
toString()...
Override Method...
Add Property...
Call Web Service Operation...
Generate REST Client...


Select Operation to Invoke
Available Web Service References:
webclient
NewWebService
NewWebService
NewWebServicePort
hello
sum



```
/**
 *
 * @author Exam
 */
public class Client {

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

    private static String sayHello() {
        server.NewWebService_Service service = new server.NewWebService_Service();
        server.NewWebService port = service.getNewWebServicePort();
        return port.sayHello();
    }

    private static Integer sum(int a, int b) {
        server.NewWebService_Service service = new server.NewWebService_Service();
        server.NewWebService port = service.getNewWebServicePort();
        return port.sum(a, b);
    }
```



```
 */
package webclient;

/**
 *
 * @author Exam
 */
public class Webclient {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        try
        {
        server.NewWebService_Service service = new server.NewWebService_Service();
        server.NewWebService port = service.getNewWebServicePort();
        System.out.println(port.hello(""));
        int result = port.sum(10, 20);
        System.out.println("Sum=" +result);
    }catch(Exception ex)
    {

    }
```