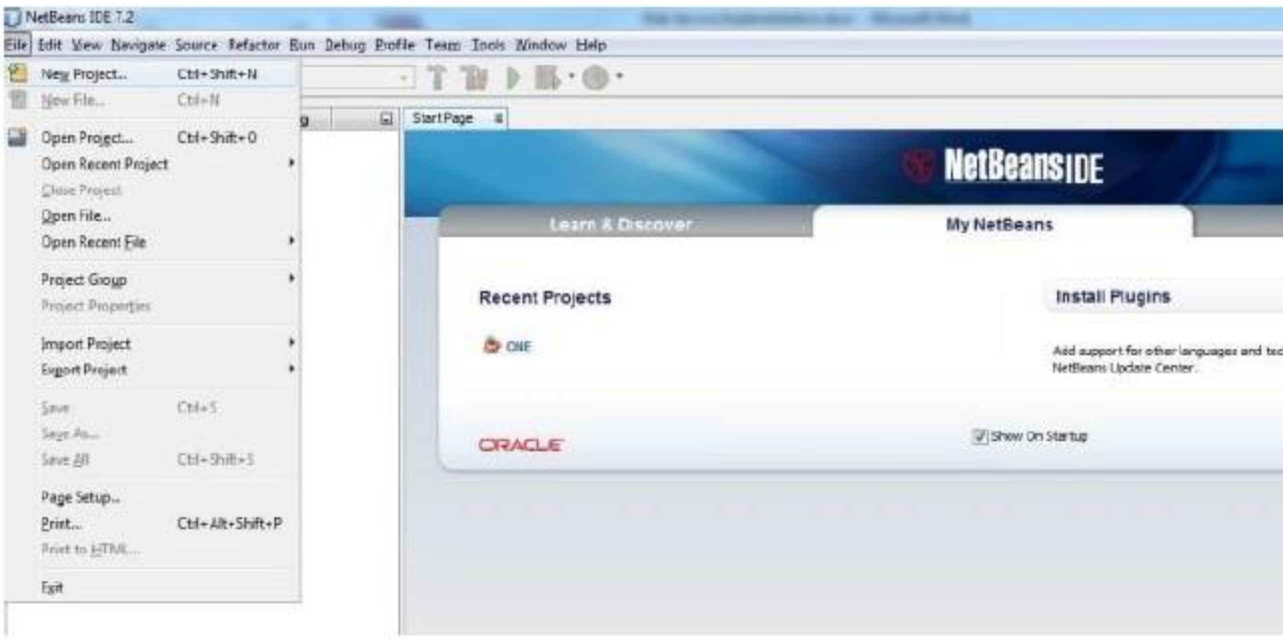
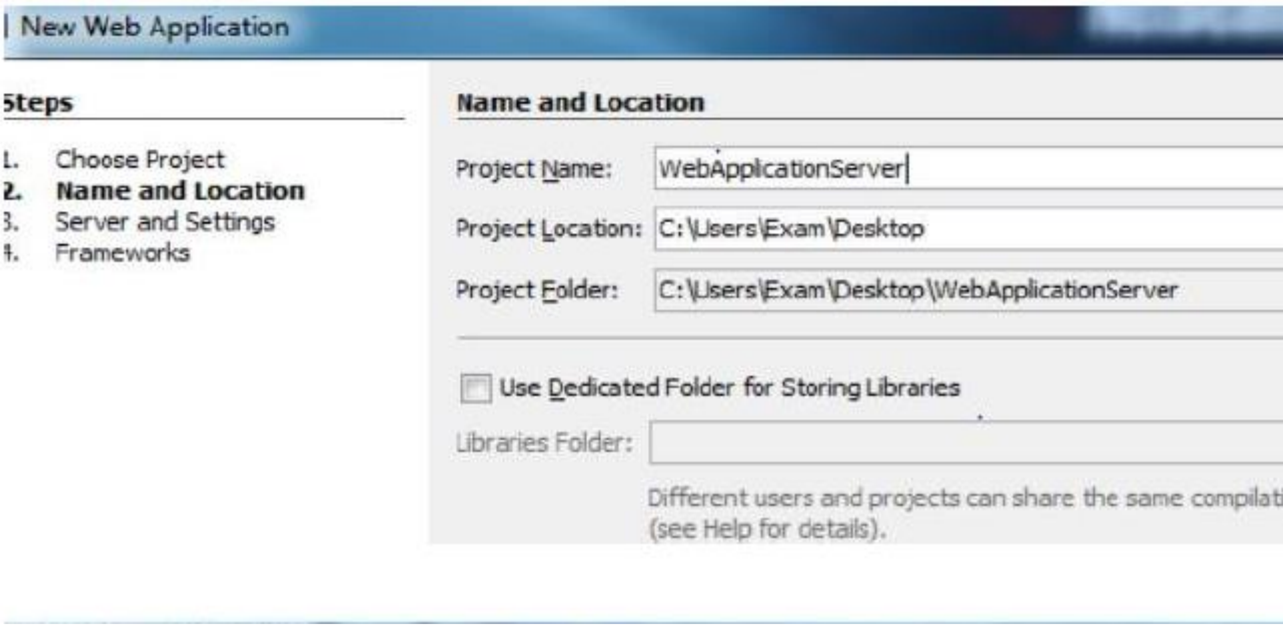
Practical – 1

AIM :Implement Web Service using java

1) Create Web Application in NetBeans
   File > New Project > Java Web > Web Application





New Web Application

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

Name and Location

Project Name: WebApplicationServer

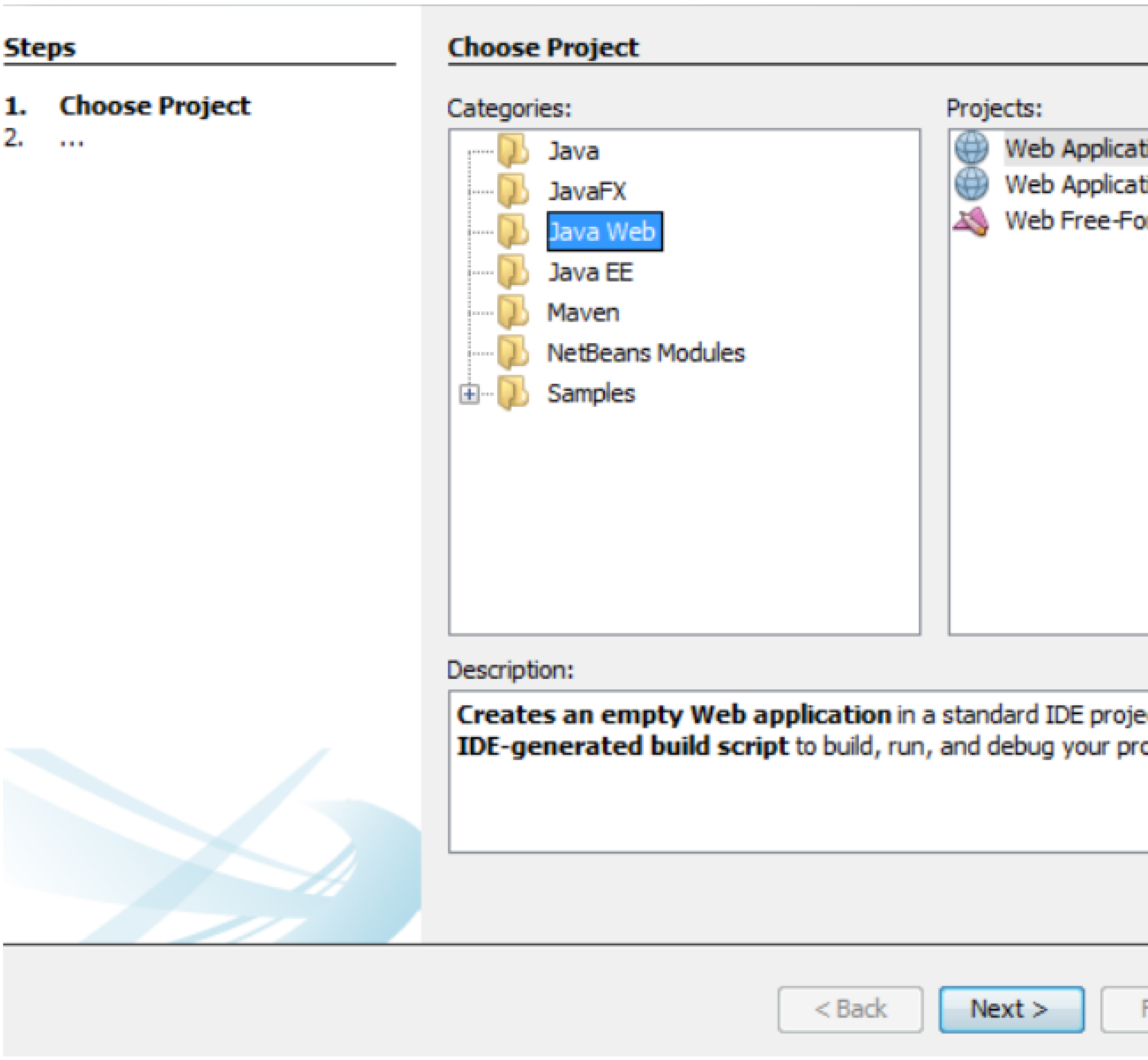
Project Location: C:\Users\Exam\Desktop

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

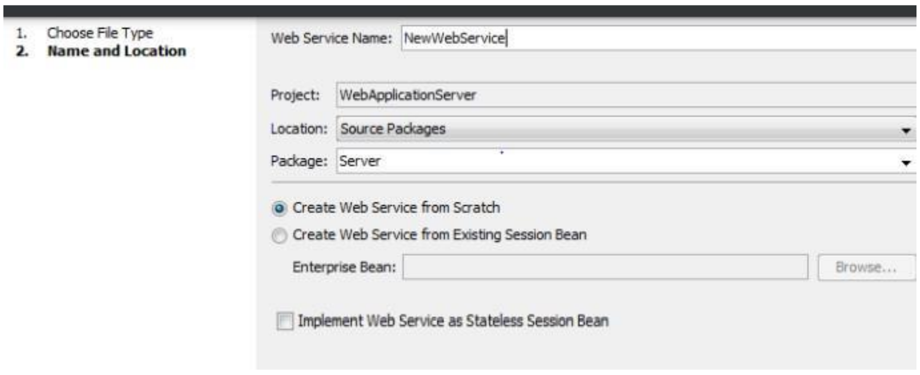
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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



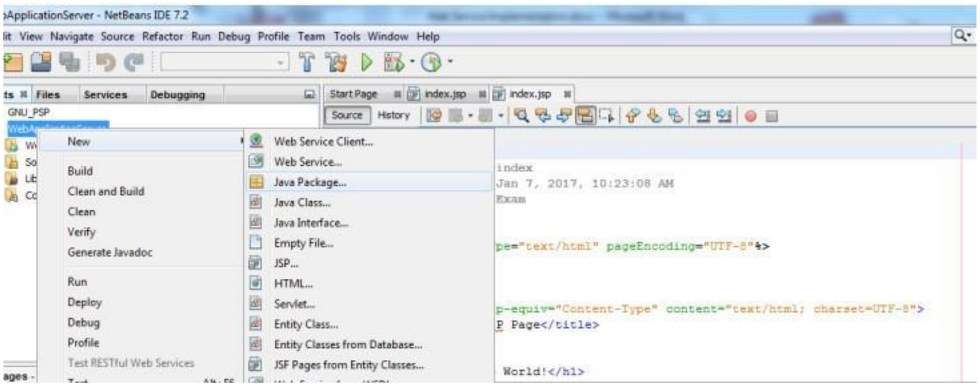
Creates an empty Web application in a standard IDE proje
IDE-generated build script to build, run, and debug your pro



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

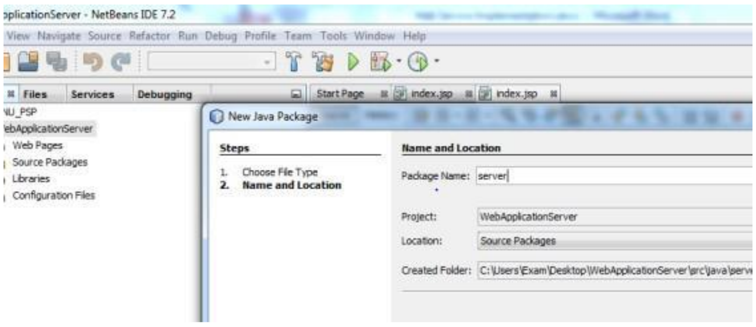
Click Next and Finish.

2) Create Java Packages from
Web Application > Source Packages > New > Java Package



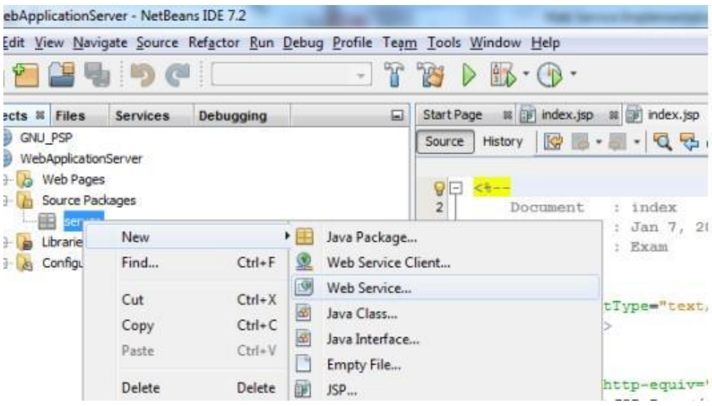
```
index
Jan 7, 2017, 10:23:08 AM
Exam

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

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

World!</h1>
```

3) Specify Name of Java Package and click on Finish Button



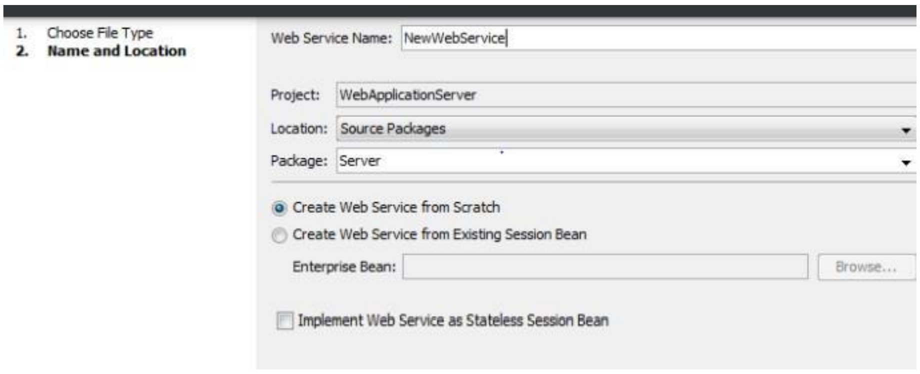
Package Name: server
Project: WebApplicationServer
Location: Source Packages
Created Folder: C:\Users\Exam\Desktop\WebApplicationServer\src\java\serv

4) Right Click on Package and create web service
New > Web Service



```
<%--
    Document   : index
               : Jan 7, 2
               : Exam
tType="text,
>
http-equiv="
```

5) Specify Name of Web service and select Java Package



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

Web Service Name: NewWebService

Project: WebApplicationServer

Location: Source Packages

Package: Server

(●) Create Web Service from Scratch

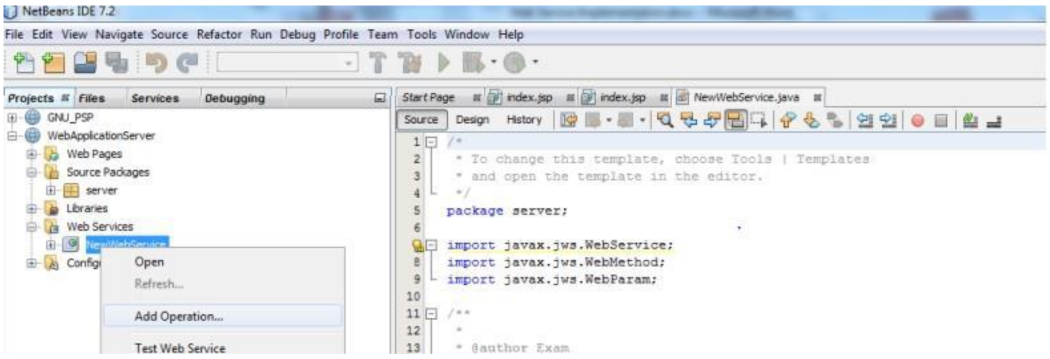
( ) Create Web Service from Existing Session Bean

Enterprise Bean:

[ ] Implement Web Service as Stateless Session Bean

Web Services Folder is created

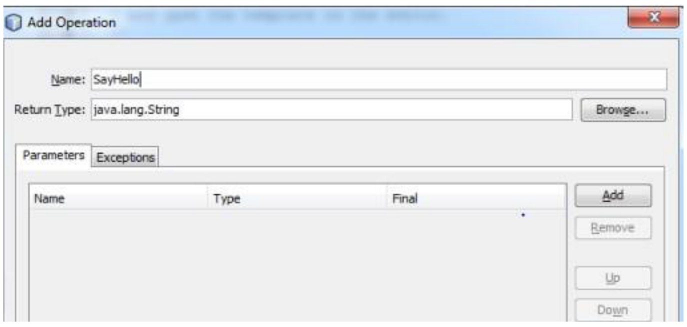
6) Under Web Services Folder Select Web service and Right click > Add Operation



```
/*
 * 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
```

7) Add Operation

Specify Name of Operation and click on OK

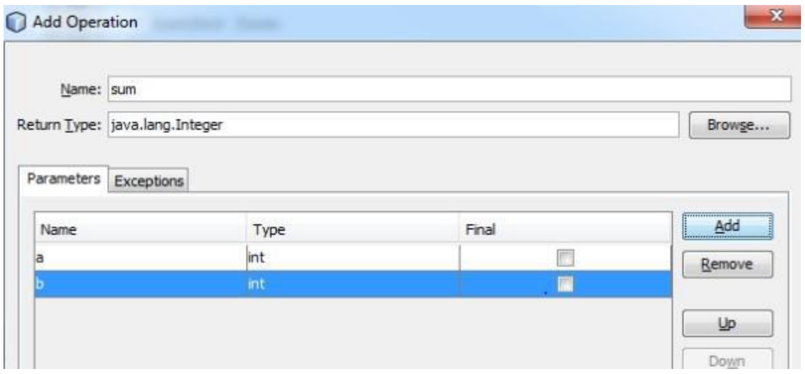


Add Operation

Name: SayHello

Return Type: java.lang.String

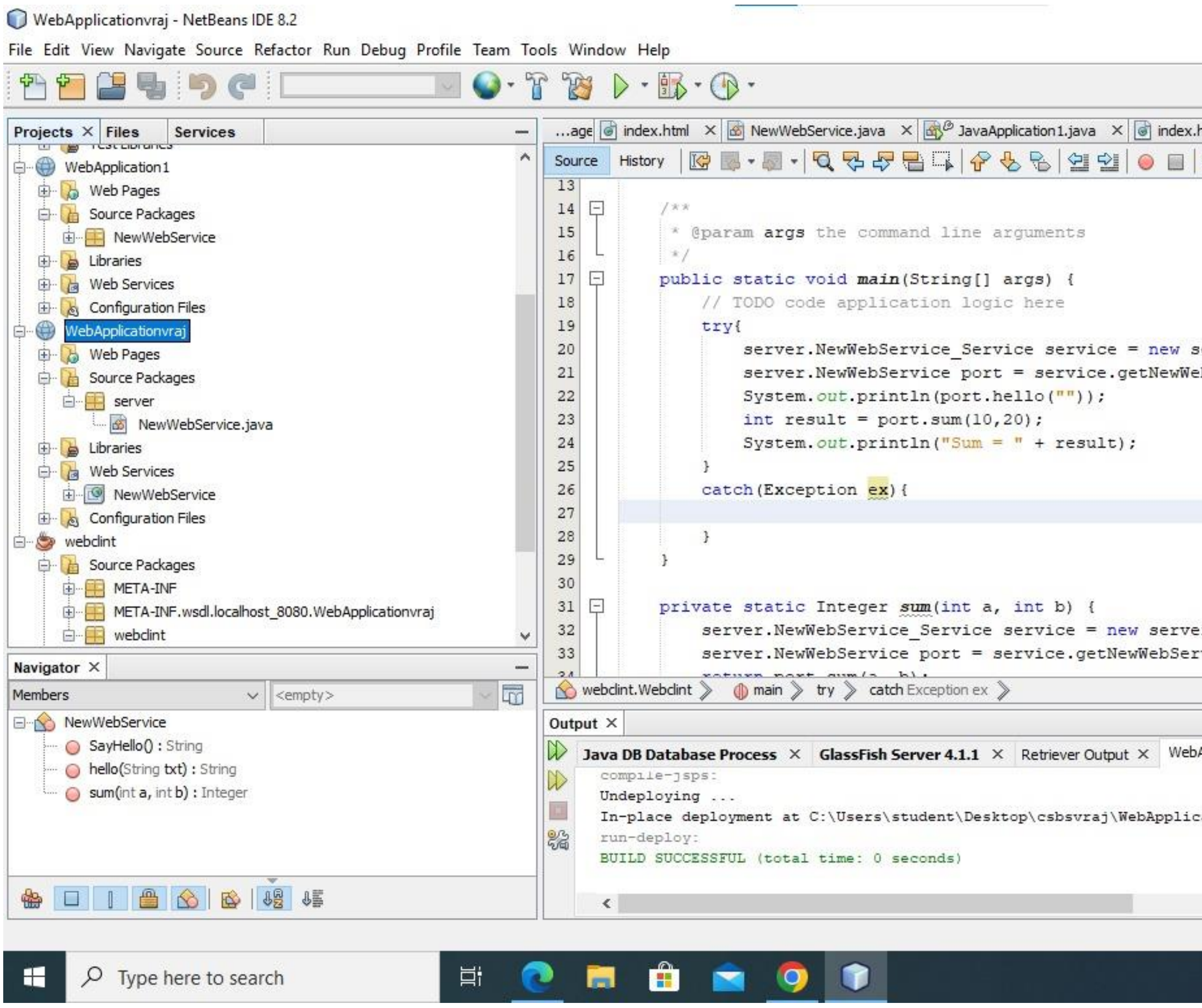
8) You May also add operation with parameters



Name: sum
Return Type: java.lang.Integer

Name	Type	Final
a	int	☐
b	int	☐

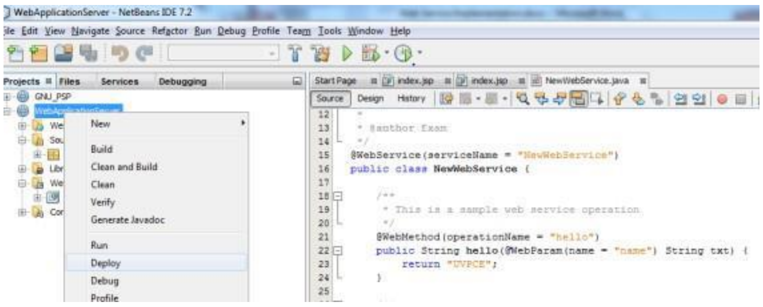
9) Modify Method according to requirement.. like I want to return string is UVPCE and perform sum operation then code


WebApplicationvraj - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
WebApplication1
Web Pages
Source Packages
NewWebService
Libraries
Web Services
Configuration Files
WebApplicationvraj
Web Pages
Source Packages
server
NewWebService.java
Libraries
Web Services
NewWebService
Configuration Files
webclint
Source Packages
META-INF
META-INF.wsdl.localhost_8080.WebApplicationvraj
webclint
Navigator
Members
<empty>
NewWebService
SayHello() : String
hello(String txt) : String
sum(int a, int b) : Integer
...age index.html NewWebService.java JavaApplication1.java index.h
Source History
webclint.Webclint main try catch Exception ex
Output
Java DB Database Process GlassFish Server 4.1.1 Retriever Output WebA
Type here to search

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

        }
    }

    private static Integer sum(int a, int b) {
        server.NewWebService_Service service = new serve
        server.NewWebService port = service.getNewWebSer
```

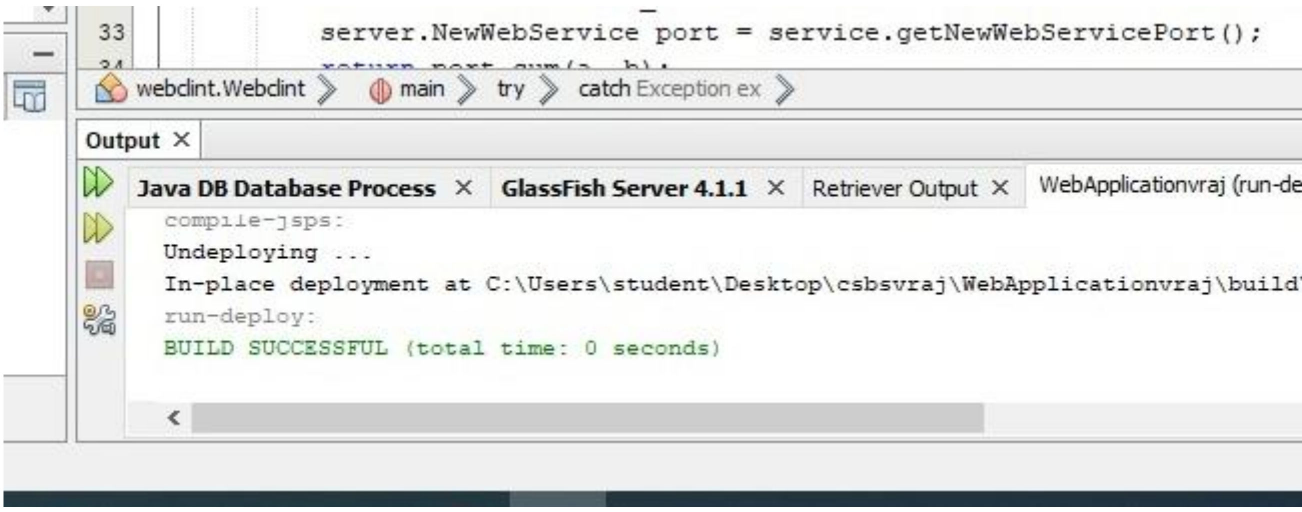
```
compile-jsps:
Undeploying ...
In-place deployment at C:\Users\student\Desktop\csbsvraj\WebApplic
run-deploy:
BUILD SUCCESSFUL (total time: 0 seconds)
```

10) Deploy Project



```
 *
 * @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";
    }
```

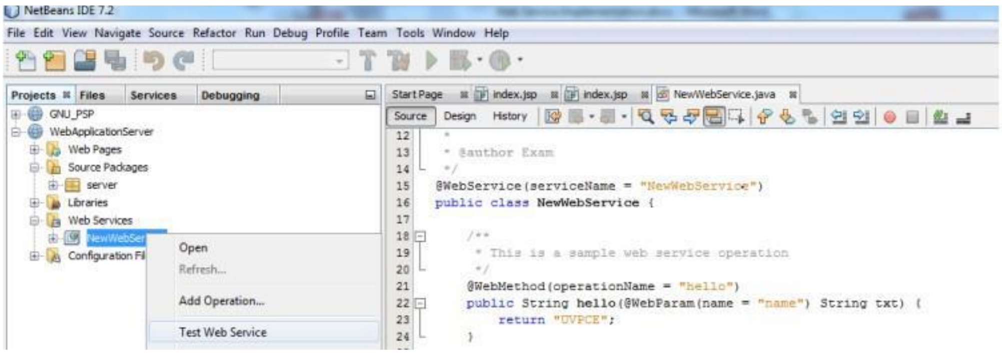
For Successful deployment



```
server.NewWebService port = service.getNewWebServicePort();
```

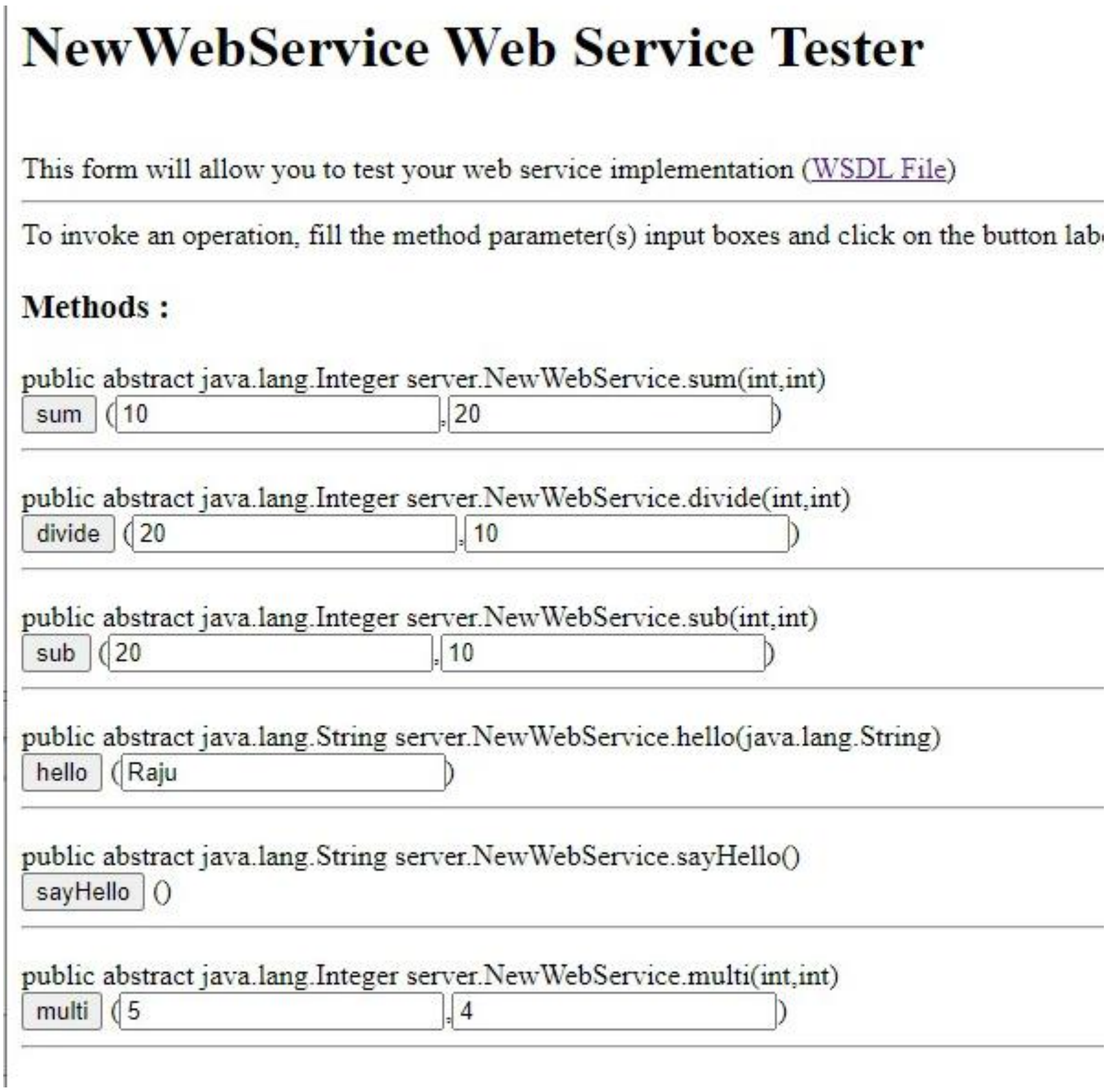
```
compile-jsps:
Undeploying ...
In-place deployment at C:\Users\student\Desktop\csbsvraj\WebApplicationvraj\build
run-deploy:
BUILD SUCCESSFUL (total time: 0 seconds)
```

11) Test Web Service



```
 *
 * @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";
    }
```

Output of Test Web service in web Browser



NewWebService 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 lab

Methods :

public abstract java.lang.Integer server.NewWebService.sum(int,int)
sum (10 , 20)

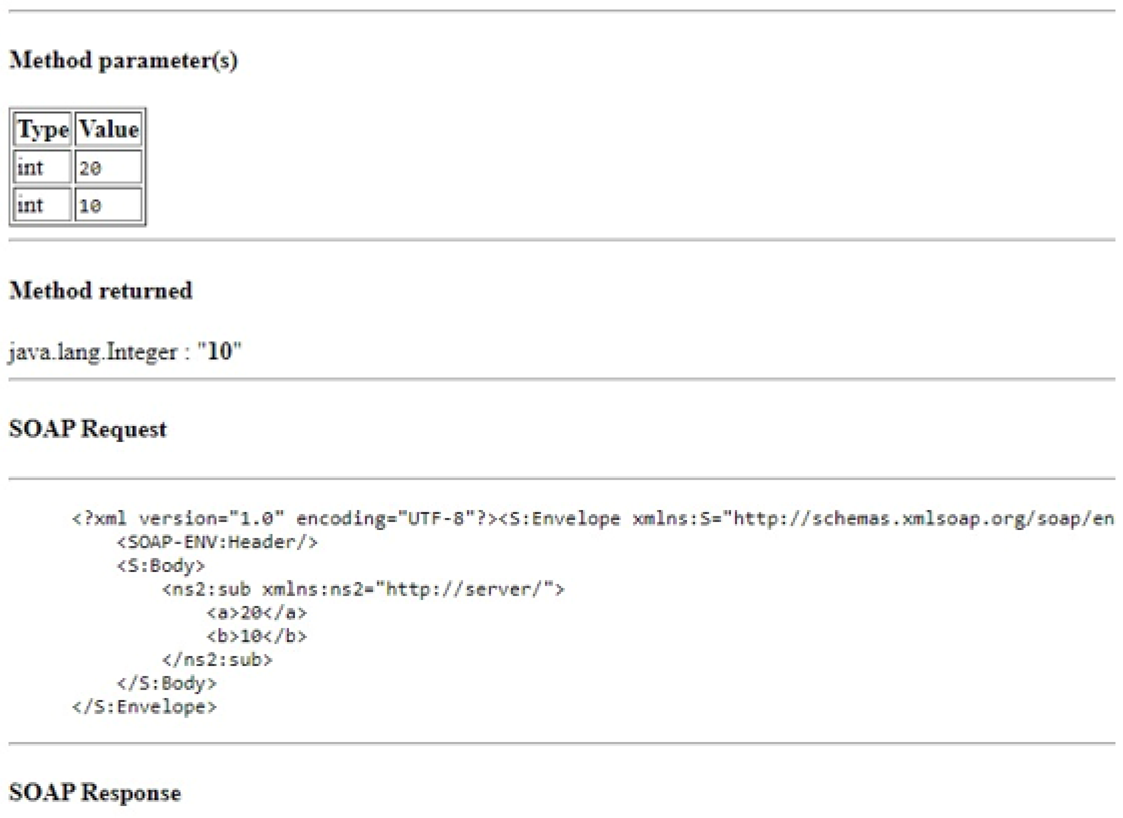
public abstract java.lang.Integer server.NewWebService.divide(int,int)
divide (20 , 10)

public abstract java.lang.Integer server.NewWebService.sub(int,int)
sub (20 , 10)

public abstract java.lang.String server.NewWebService.hello(java.lang.String)
hello (Raju)

public abstract java.lang.String server.NewWebService.sayHello()
sayHello ()

public abstract java.lang.Integer server.NewWebService.multi(int,int)
multi (5 , 4)



Method parameter(s)

Type	Value
int	20
int	10

Method returned

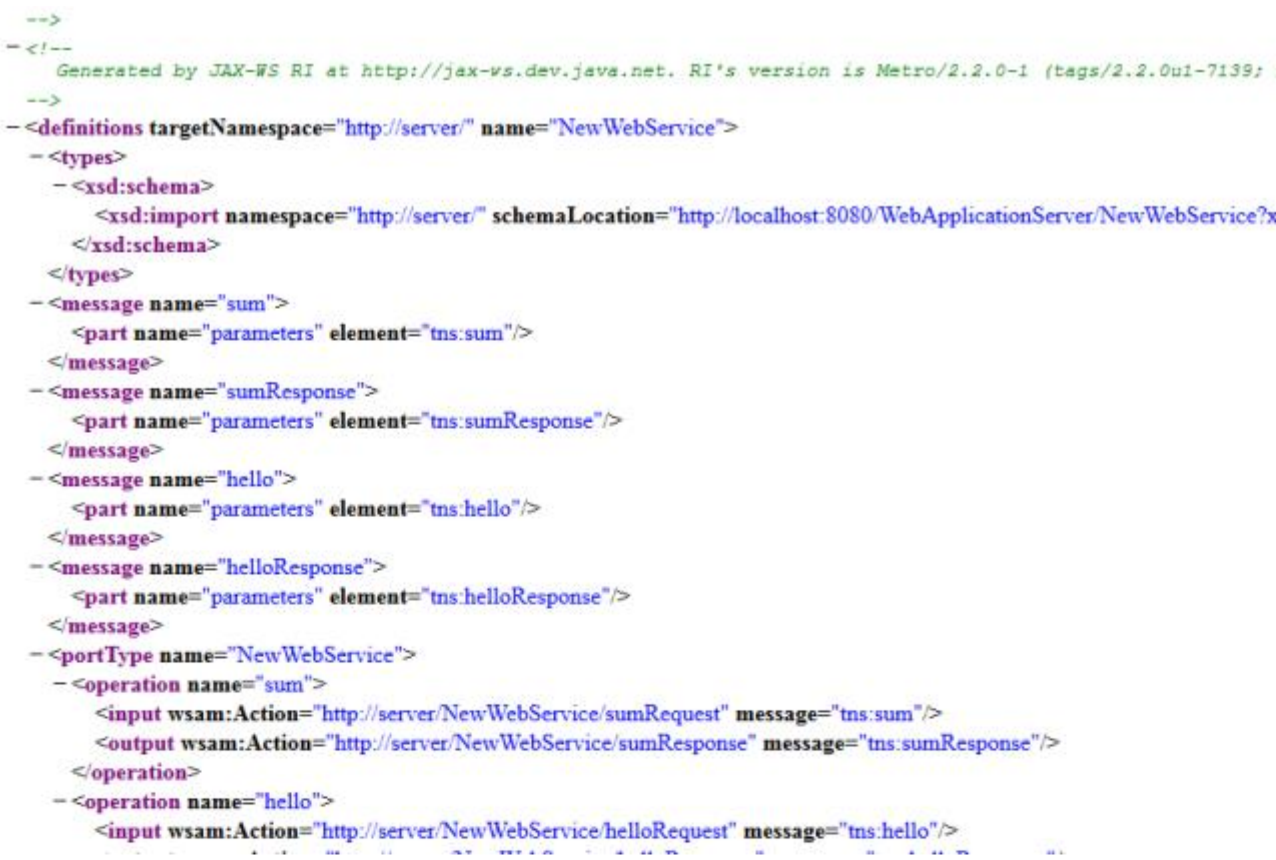
java.lang.Integer : "10"

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:sub xmlns:ns2="http://server/">
            <a>20</a>
            <b>10</b>
        </ns2:sub>
    </S:Body>
</S:Envelope>
```

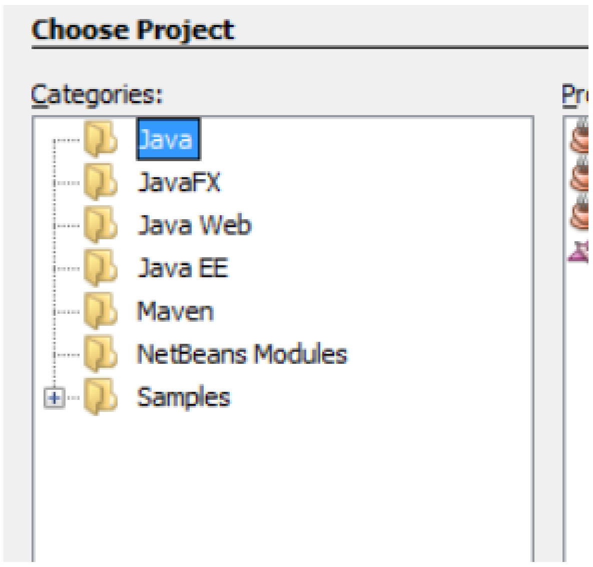
SOAP Response

12) Click on WSDL File Link and copy newly open window URL

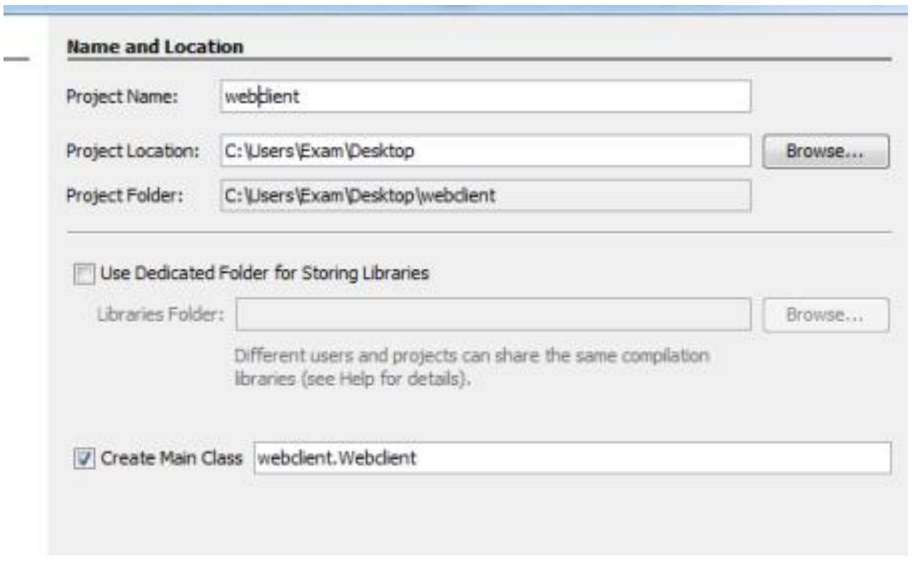


```
   -->
 - <!--
     Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.2.0-1 (tags/2.2.0u1-7139;
   -->
 - <definitions targetNamespace="http://server/" name="NewWebService">
   - <types>
     - <xsd:schema>
         <xsd:import namespace="http://server/" schemaLocation="http://localhost:8080/WebApplicationServer/NewWebService?x
       </xsd:schema>
     </types>
   - <message name="sum">
       <part name="parameters" element="tns:sum"/>
     </message>
   - <message name="sumResponse">
       <part name="parameters" element="tns:sumResponse"/>
     </message>
   - <message name="hello">
       <part name="parameters" element="tns:hello"/>
     </message>
   - <message name="helloResponse">
       <part name="parameters" element="tns:helloResponse"/>
     </message>
   - <portType name="NewWebService">
     - <operation name="sum">
         <input wsam:Action="http://server/NewWebService/sumRequest" message="tns:sum"/>
         <output wsam:Action="http://server/NewWebService/sumResponse" message="tns:sumResponse"/>
       </operation>
     - <operation name="hello">
         <input wsam:Action="http://server/NewWebService/helloRequest" message="tns:hello"/>
```

13) Create Simple Java Application from File > New > Java > java Application



14) Specify Name and Finish



Name and Location

Project Name: webclient

Project Location: C:\Users\Exam\Desktop

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

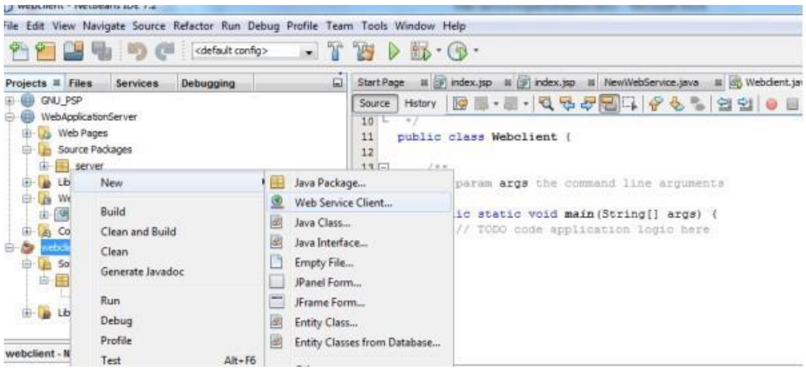
[ ] Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

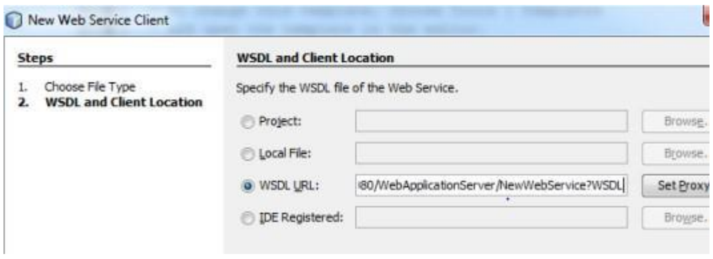
[✓] Create Main Class webclient.Webclient

15) Add Web Service Client from
Client Project > Source Package > New > Web Service Client



```
public class Webclient {

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

16) Add WSDL and click on Finish



WSDL and Client Location

Specify the WSDL file of the Web Service.

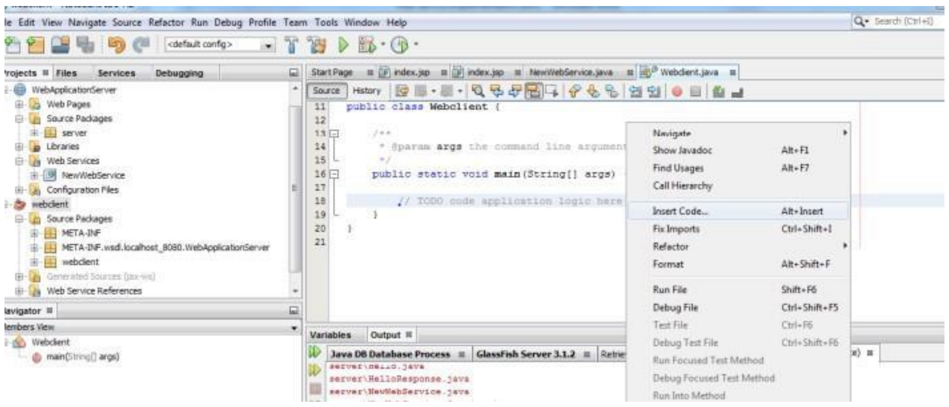
( ) Project:

( ) Local File:

(●) WSDL URL: 80/WebApplicationServer/NewWebService?WSDL

( ) IDE Registered:

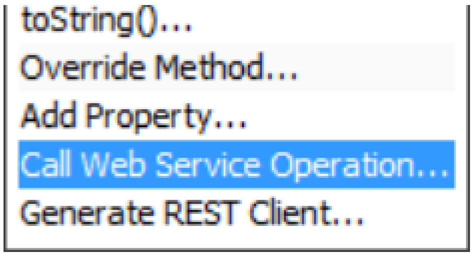
17) Open code of java file and Right click > Insert Code



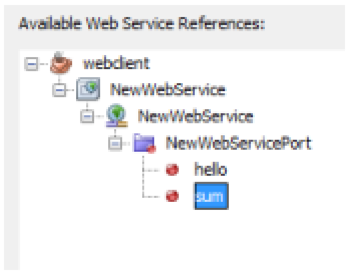
```
public class Webclient {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        // TODO code application logic here
    }
}
```

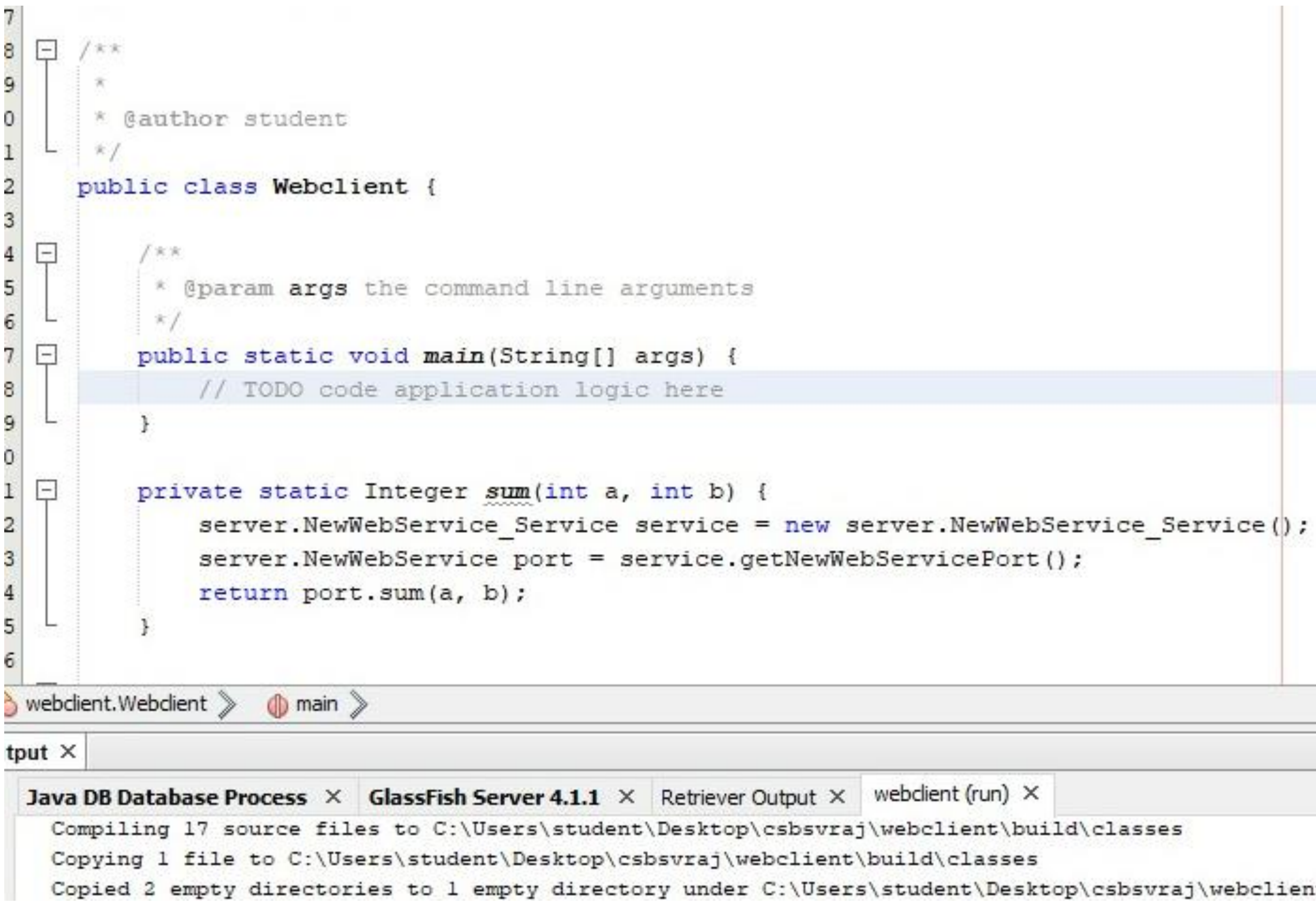
```
server\Hello.java
server\HelloResponse.java
server\NewWebService.java
```

Then Select Call Web Service Operation



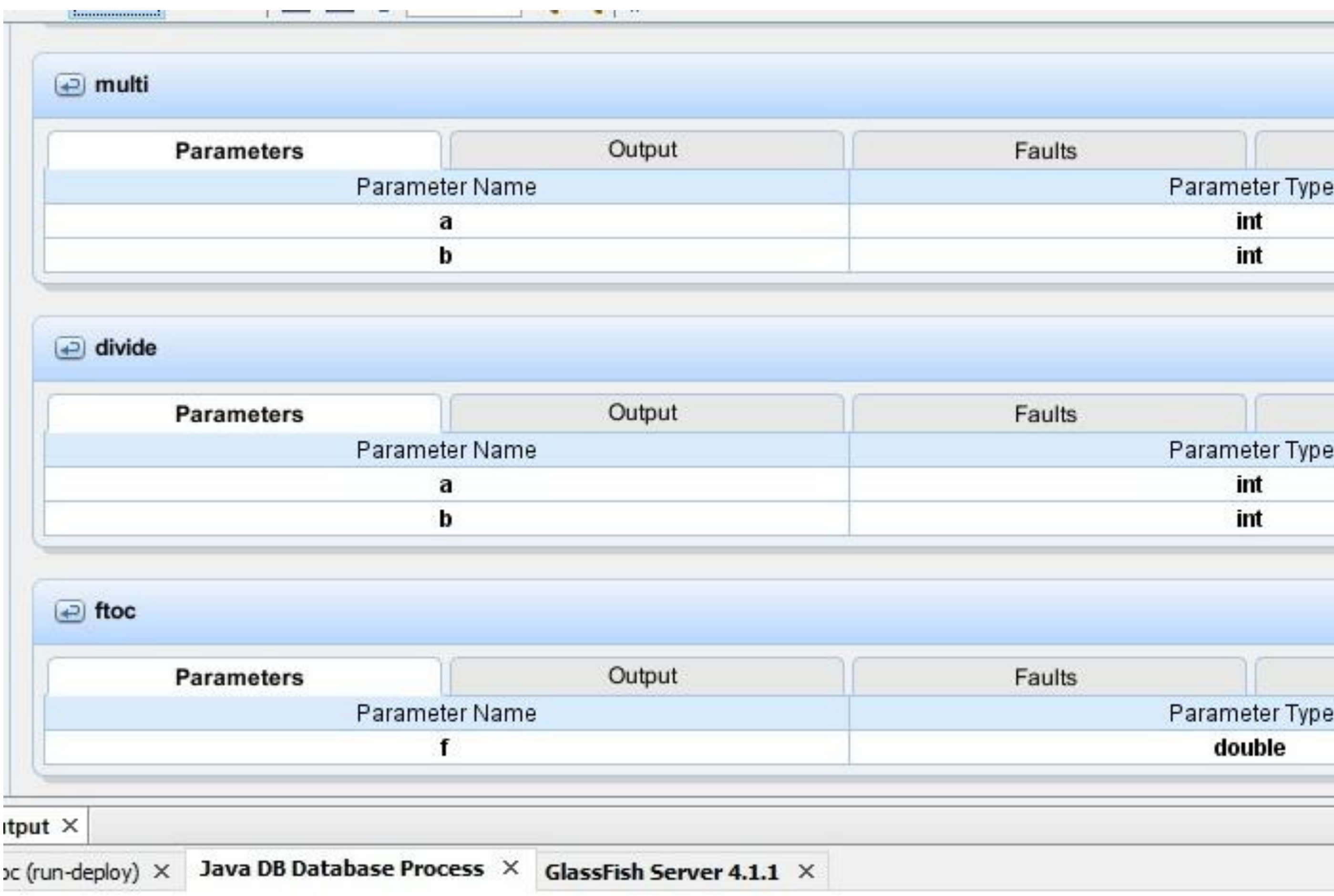
18) Select Operation from Web Service





```
/**
 *
 * @author student
 */
public class Webclient {

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

    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);
    }
```

```
Compiling 17 source files to C:\Users\student\Desktop\csbsvraj\webclient\build\classes
Copying 1 file to C:\Users\student\Desktop\csbsvraj\webclient\build\classes
Copied 2 empty directories to 1 empty directory under C:\Users\student\Desktop\csbsvraj\webclien
```



multi

Parameters	Output	Faults

Parameter Name	Parameter Type
a	int
b	int

divide

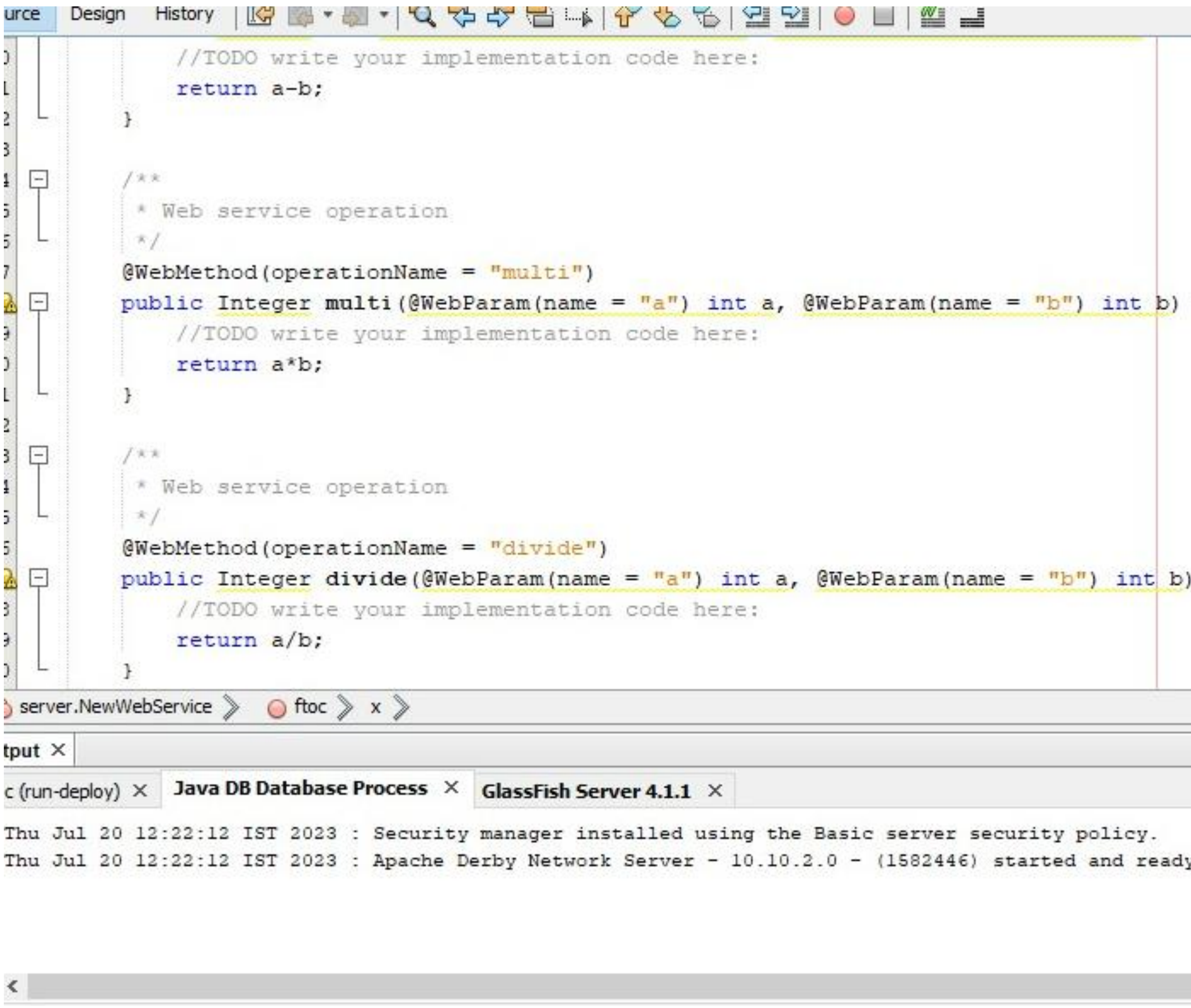
Parameters	Output	Faults

Parameter Name	Parameter Type
a	int
b	int

ftoc

Parameters	Output	Faults

Parameter Name	Parameter Type
f	double



```
        //TODO write your implementation code here:
        return a-b;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "multi")
    public Integer multi(@WebParam(name = "a") int a, @WebParam(name = "b") int b)
        //TODO write your implementation code here:
        return a*b;
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "divide")
    public Integer divide(@WebParam(name = "a") int a, @WebParam(name = "b") int b)
        //TODO write your implementation code here:
        return a/b;
    }
```

```
Thu Jul 20 12:22:12 IST 2023 : Security manager installed using the Basic server security policy.
Thu Jul 20 12:22:12 IST 2023 : Apache Derby Network Server - 10.10.2.0 - (1582446) started and ready
```