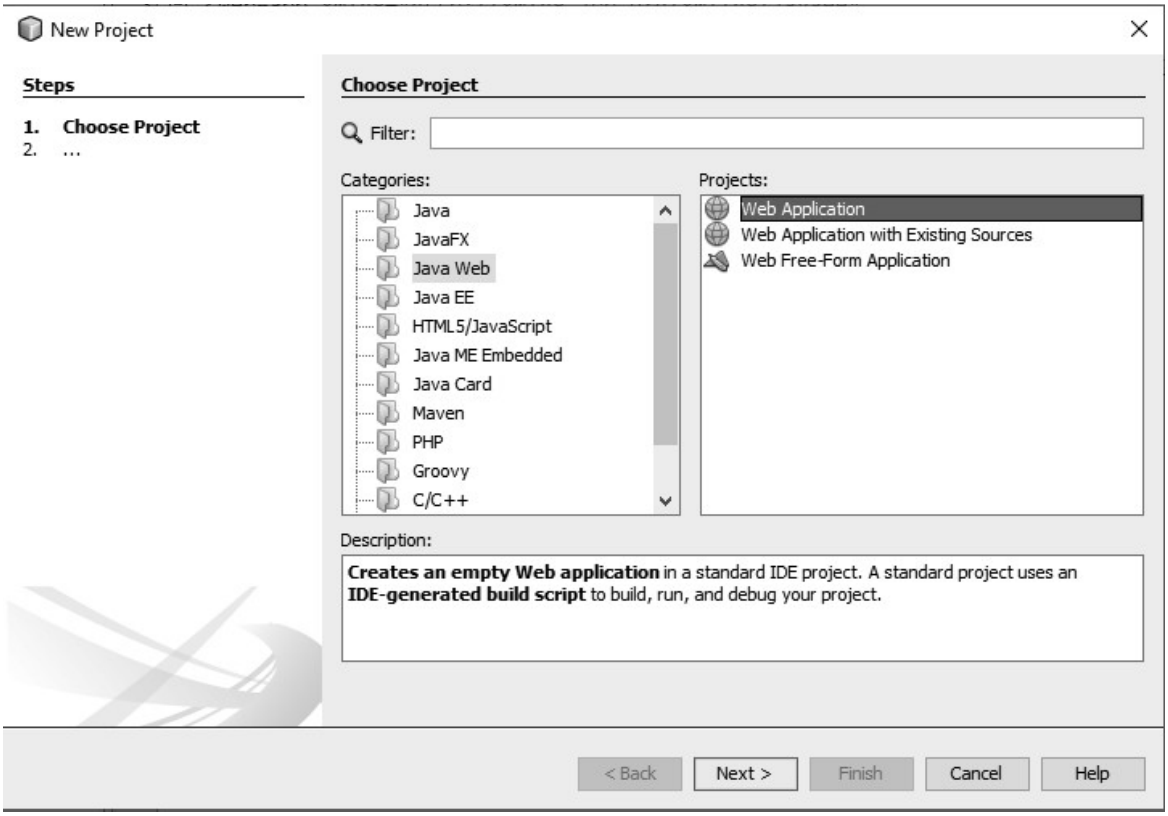
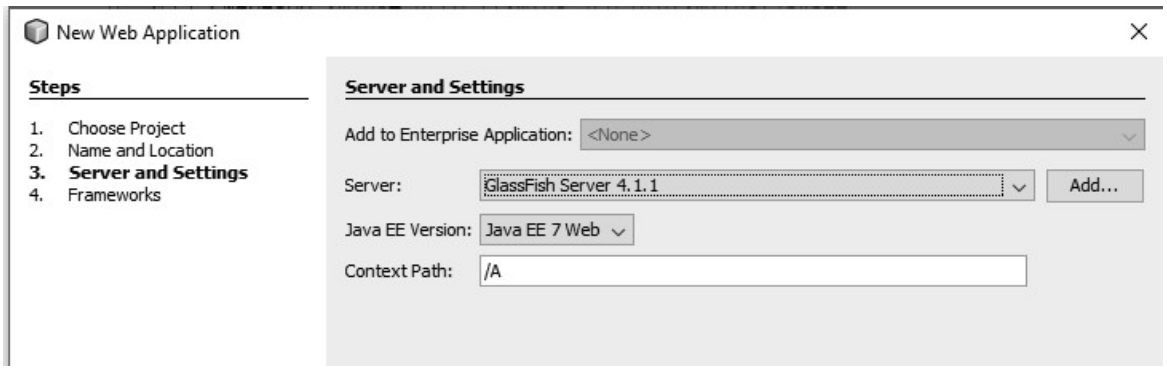
Step 1 : Create a new project

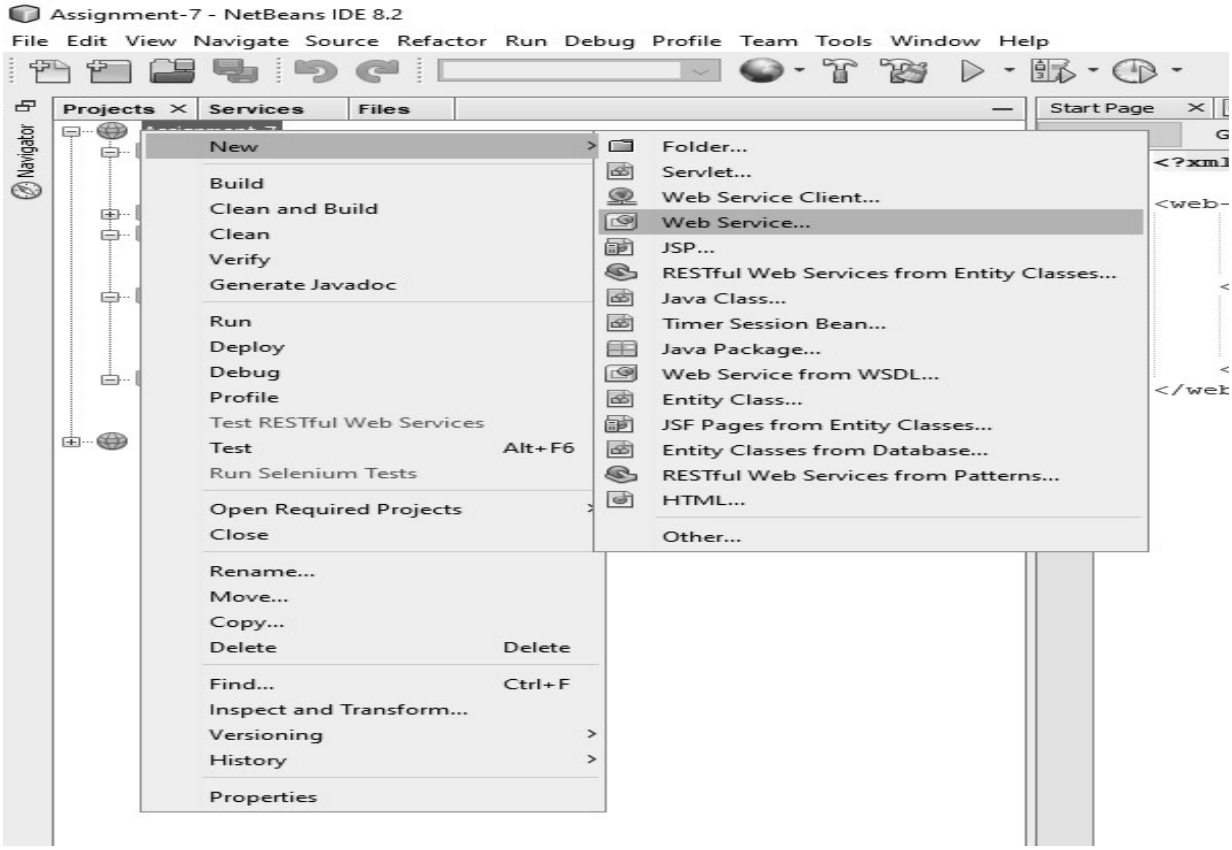
File -> New Project -> Java Web -> Web Application ->Next ->Enter Project Name (Assignment 7)-> Next -> Finish





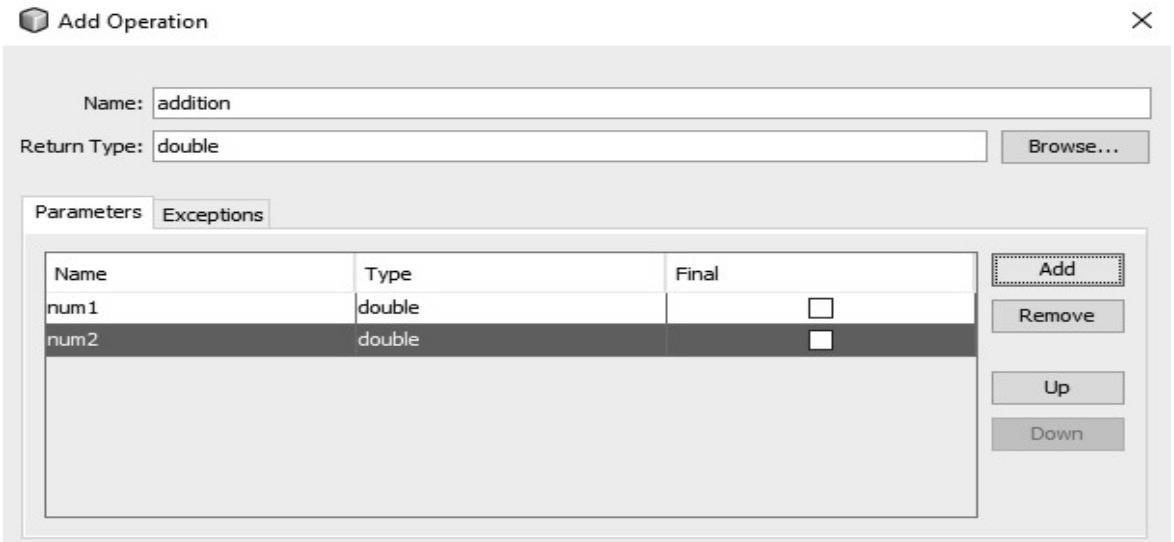
Step 2 : Create a new Web Services

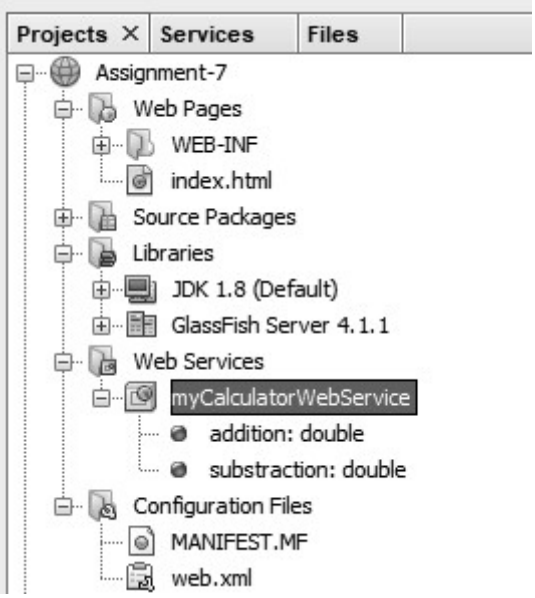
Right click on project name -> New -> Web service -> Enter name (MyCalculatorWebService) -> in package field enter (com.myservice) -> Finish



Step 3 : Add Operations

Right click on MyCalculatorWebService -> Add Operation -> Enter name (addition, subtraction, etc) -> Enter return type (double) -> Add parameters (num1, num2) ->OK





Change the function return 0.0 to num1 (operation) num2 for all operations.
Example: num1+num**2**

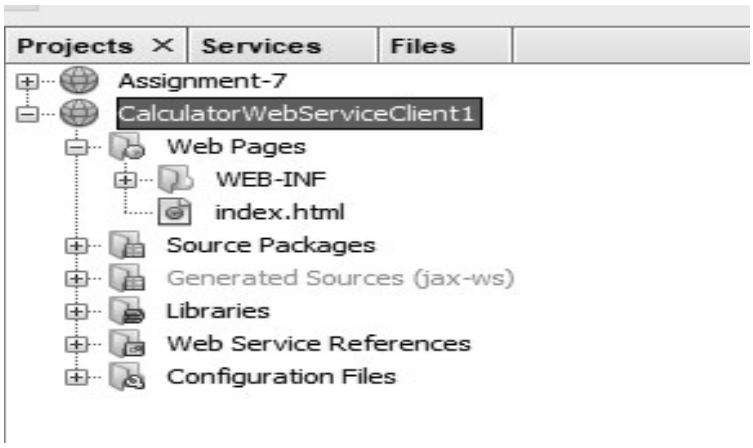
```
15      */
16      @WebService(serviceName = "myCalculatorWebService")
17      public class myCalculatorWebService {
18
19          /**
20           * Web service operation
21           */
22          @WebMethod(operationName = "addition")
            public double addition(@WebParam(name = "num1") double num1, @WebParam(name = "num2") double num2) {
24              //TODO write your implementation code here:
25              return num1 + num2;
26          }
27
28          /**
29           * Web service operation
30           */
31          @WebMethod(operationName = "substraction")
            public double substraction(@WebParam(name = "num1") double num1, @WebParam(name = "num2") double num2) {
33              //TODO write your implementation code here:
34              return num1 - num2;
35          }
36
```

Step 4 : Build Project

Right click on project -> Build

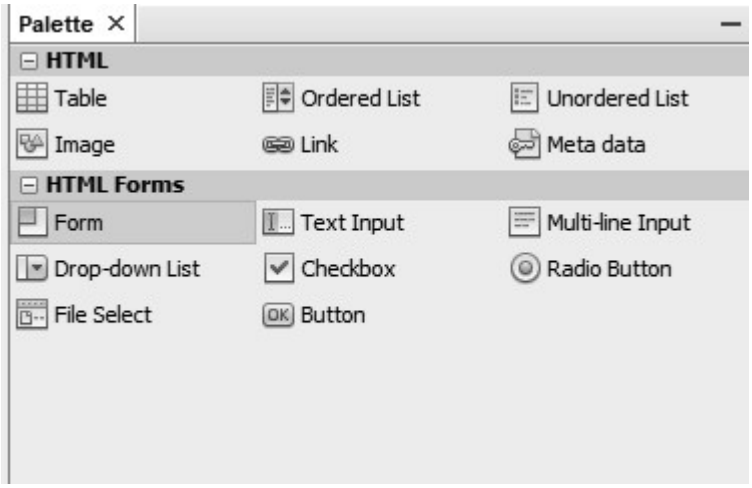
Step 5 : Create new Project

File -> New Project -> Java Web -> Web Application ->Next ->Enter Project Name (CalculatorWebServiceClient1)-> Next -> Finish

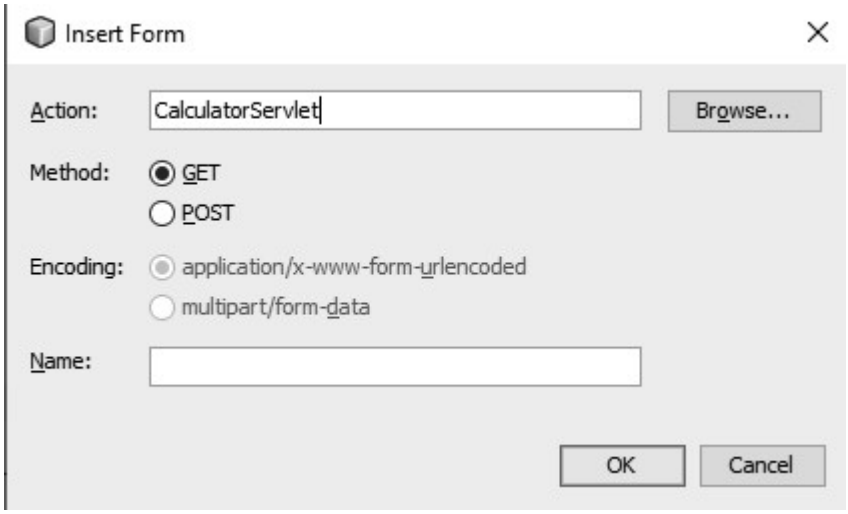


Step 6 : Create index page

Windows -> IDE Tools -> Palette -> Add required form, text fields and buttons.



1) Drag and drop Form in the body section -> Enter Action (Calculatorservlet) -> Method: Get -> OK



2) Add the text field for numbers input from the palette.

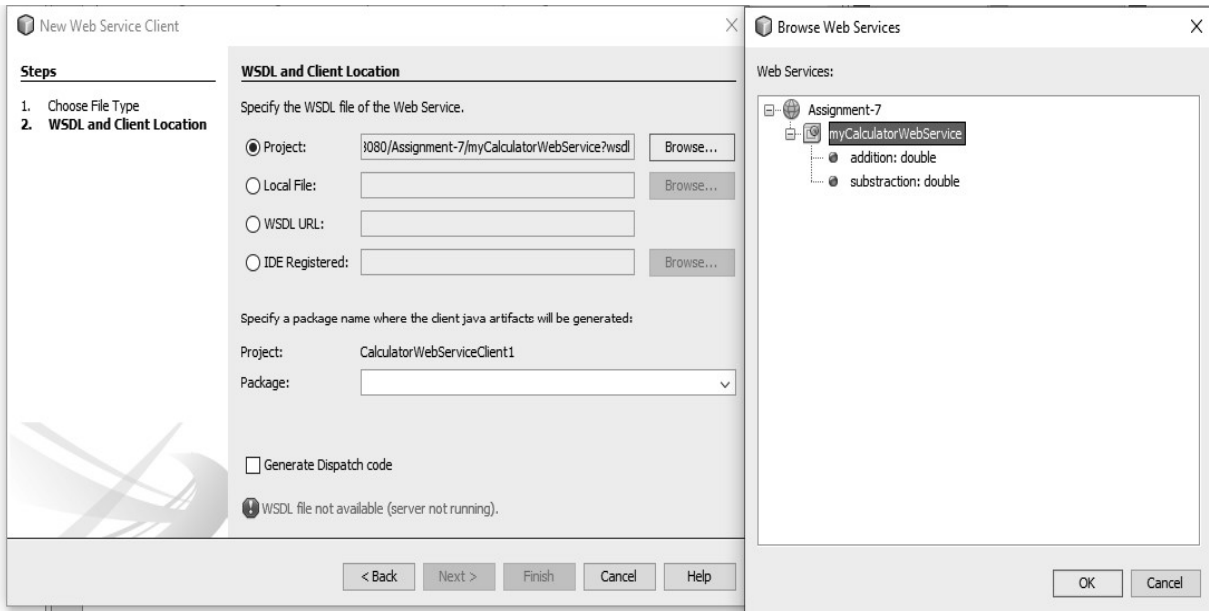
index.html | CalculatorServlet.java

Source | History

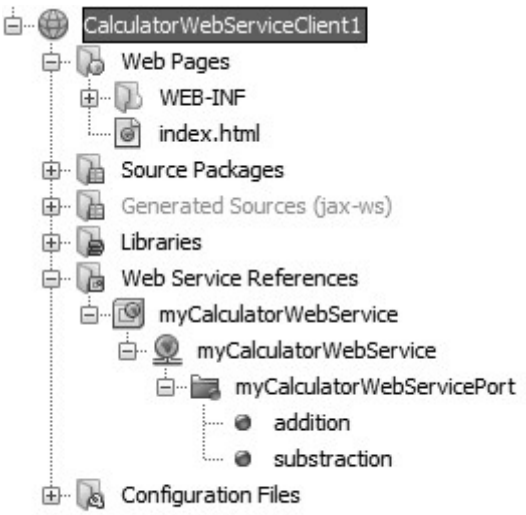
```
1  <!DOCTYPE html>
2  <!--
3  To change this license header, choose License Headers in Project Properties.
4  To change this template file, choose Tools | Templates
5  and open the template in the editor.
6  -->
7  <html>
8      <head>
9          <title>calculator web Service Client</title>
10         <meta charset="UTF-8">
11         <meta name="viewport" content="width=device-width, initial-scale=1.0">
12     </head>
13     <body>
14         <form action="CalculatorServlet">
15             Enter number 1 : <input type="text" name="number1" value="" /><br>
16             Enter number 2 : <input type="text" name="number2" value="" /><br>
17
18             <input type="submit" value="Submit" />
19         </form>
20
21
22     </body>
</html>
24
```

Step 7 : Create new Web Servlet Client

Right click on CalculatorWebServiceClient1-> New -> Web Service Client -> Browse the Project Assignment 7 -> OK -> Finish.

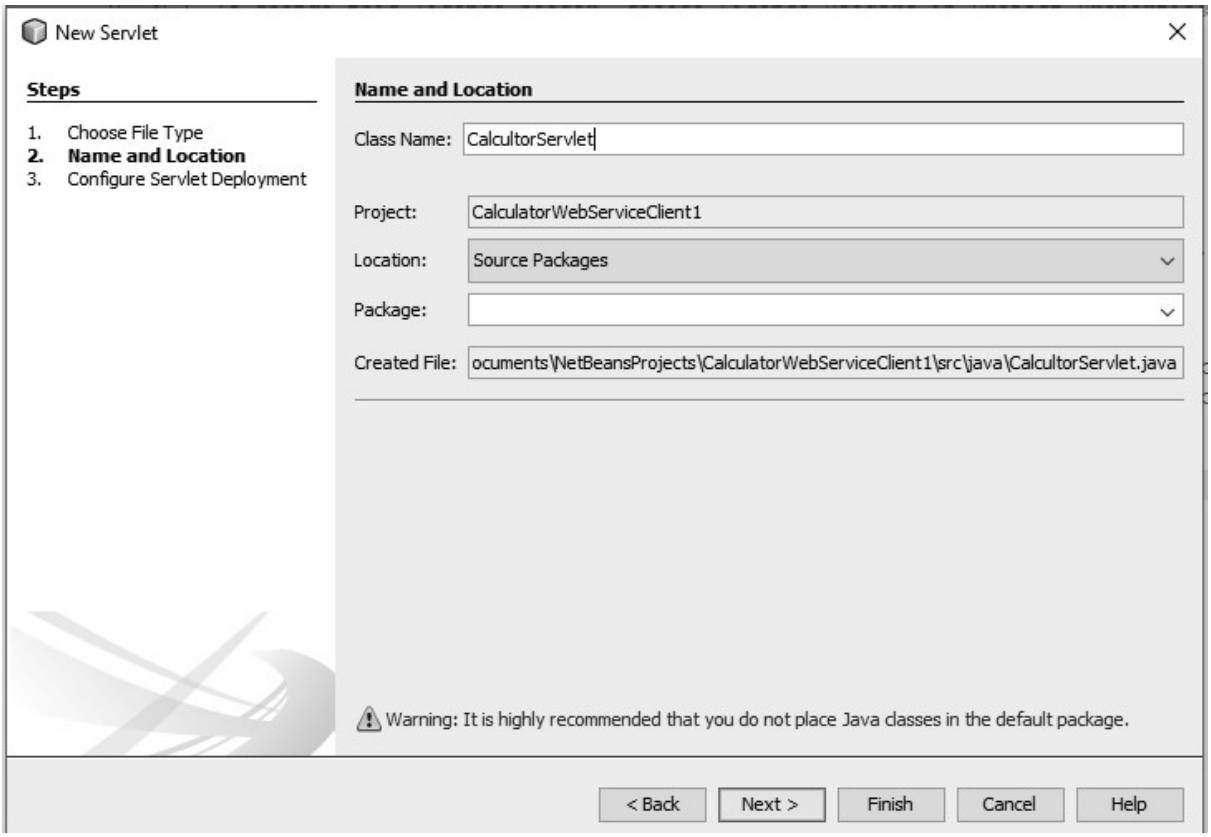


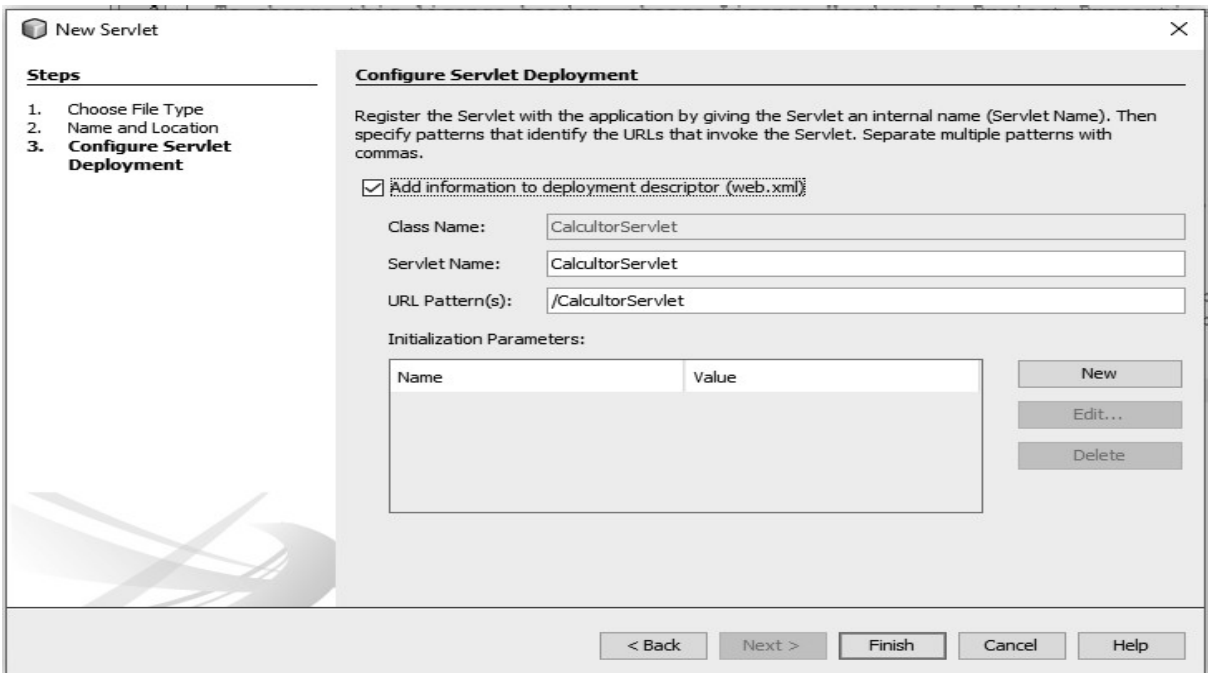
Web Service References are created successfully.



Step 8 : Create new Servlet

Right click on CalculatorWebServiceClient1-> New -> Service…-> Enter class name (CalculatorServlet) -> Next -> Check the box for adding information to web.xml -> Finish.





Step 9 : Add Operations to the CalculatorServlet

Drag and drop each operations to the CalculatorServlet.java file

```
    HttpServlet methods. Click on the + sign on the left to edit the code.

    private double addition(double num1, double num2) {
        // Note that the injected javax.xml.ws.Service reference as well as port objects are not thread safe.
        // If the calling of port operations may lead to race condition some synchronization is required.
        com.myservice.MyCalculatorWebService port = service.getMyCalculatorWebServicePort();
        return port.addition(num1, num2);
    }

    private double substraction(double num1, double num2) {
        // Note that the injected javax.xml.ws.Service reference as well as port objects are not thread safe.
        // If the calling of port operations may lead to race condition some synchronization is required.
        com.myservice.MyCalculatorWebService port = service.getMyCalculatorWebServicePort();
        return port.substraction(num1, num2);
    }

}
```

Step 10 : Store the numbers in CalculatorServlet

Numbers defined in index.html file need to be stored in the Servlet to access.

```
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {

            double num1, num2;

            num1 = Double.parseDouble(request.getParameter("number1"));
            num2 = Double.parseDouble(request.getParameter("number2"));
```

Step 11 : Change the output to be displayed

Results in the format for operations like addition, subtraction, etc

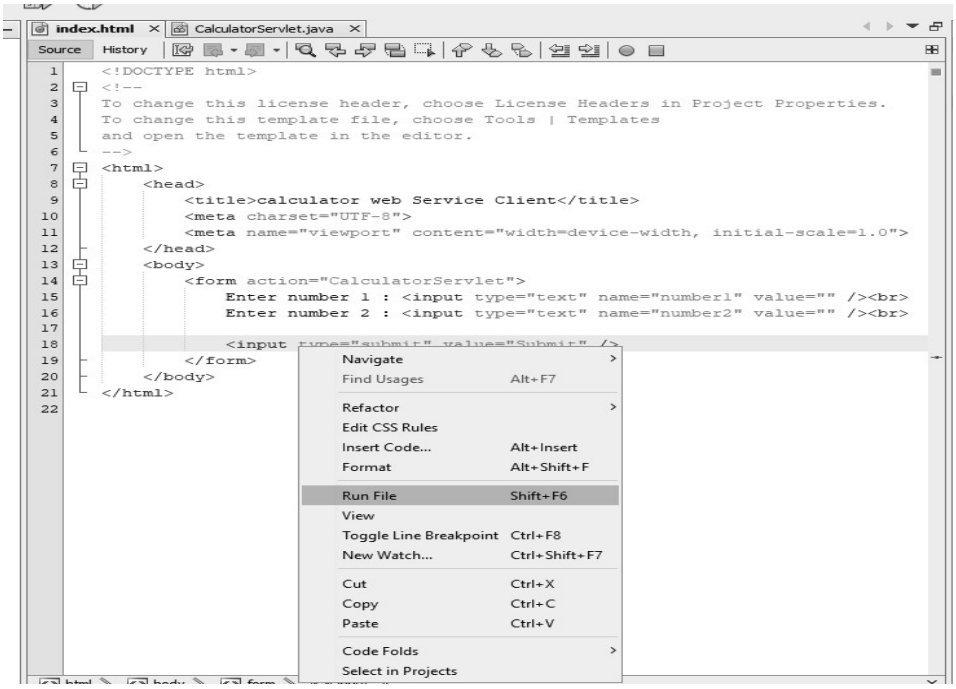
```
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet CalculatorServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Addition is  " + addition(num1, num2) + "</h1>");
            out.println("<h1>Substraction is  " + substraction(num1, num2) + "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }
```

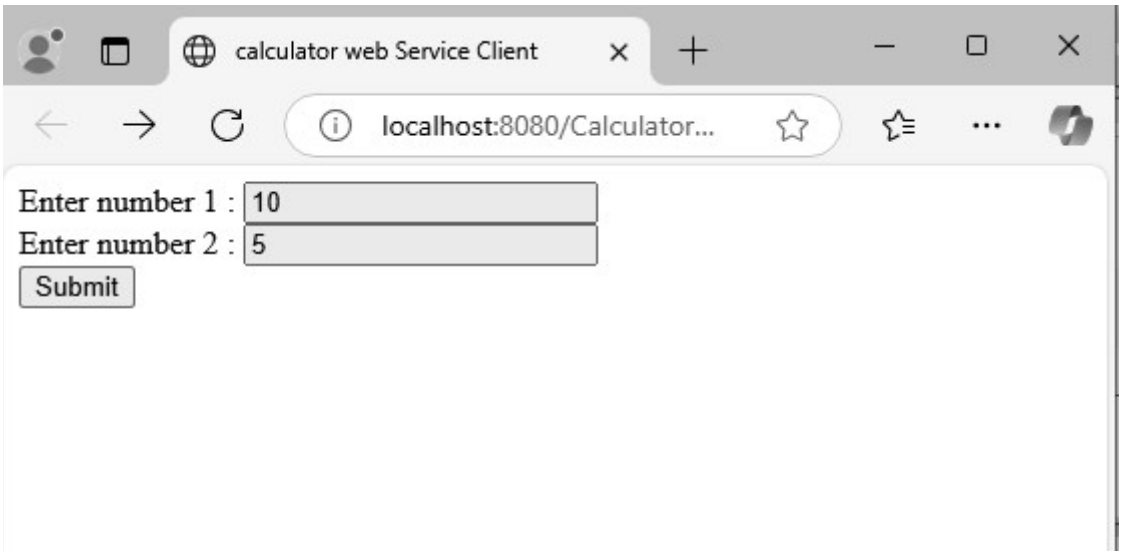
Step 12 : Clean and Build

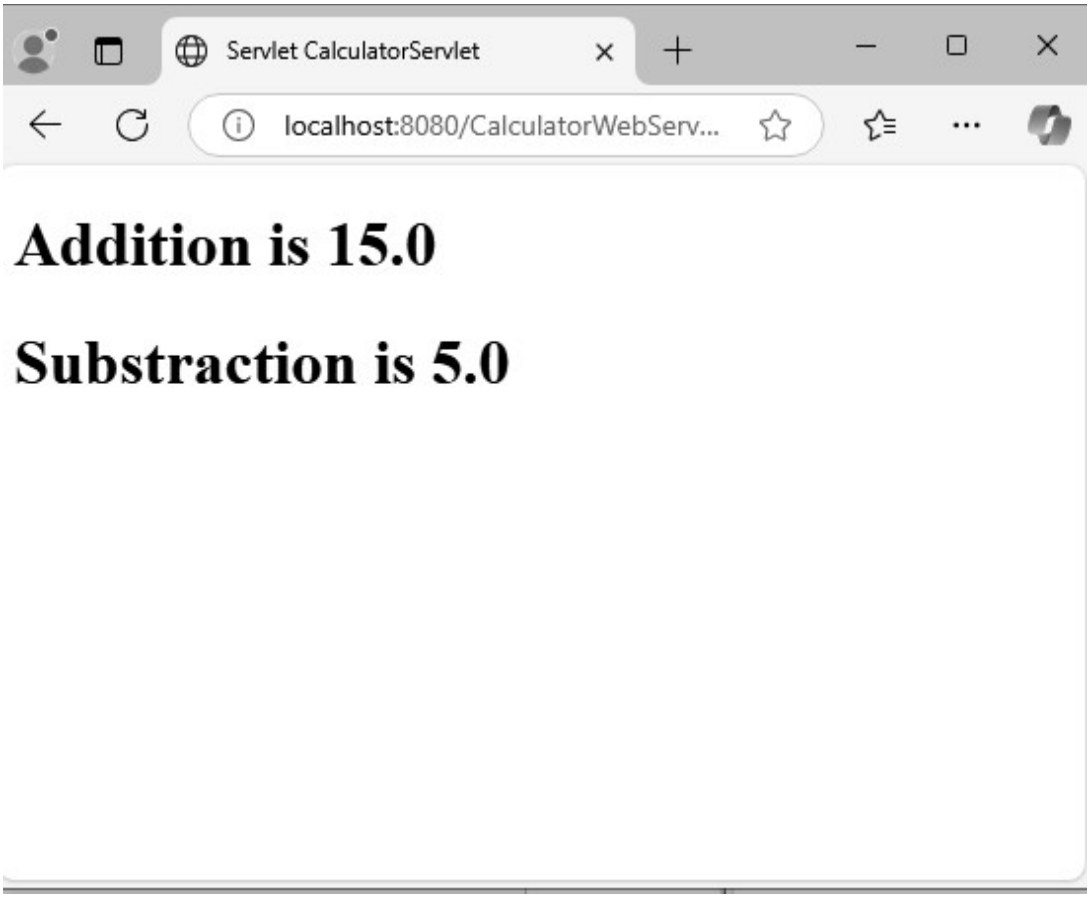
Right click on CalculatorWebServiceClient1-> Clean and Build

Step 13 : Run File

CalculatorWebServiceClient1 ->Web Pages -> index.html -> Right click somewhere on the file -> Run File




calculator web Service Client
localhost:8080/Calculator...
Enter number 1 : 10
Enter number 2 : 5
Submit


Servlet CalculatorServlet
localhost:8080/CalculatorWebServ...
Addition is 15.0
Substraction is 5.0