

# LAB 10: Enterprise Java Bean Lab Activities

---

## Exercise 1: Creating a new EJB module

Task 1: Create a new project for ejb module

In the New Project wizard, expand the **Java EE category** and select **EJB Module** as shown in the figure below. Then click Next.

- a) Create the EJB module project with the following characteristic: Project:

BasicOperationsEJBModule

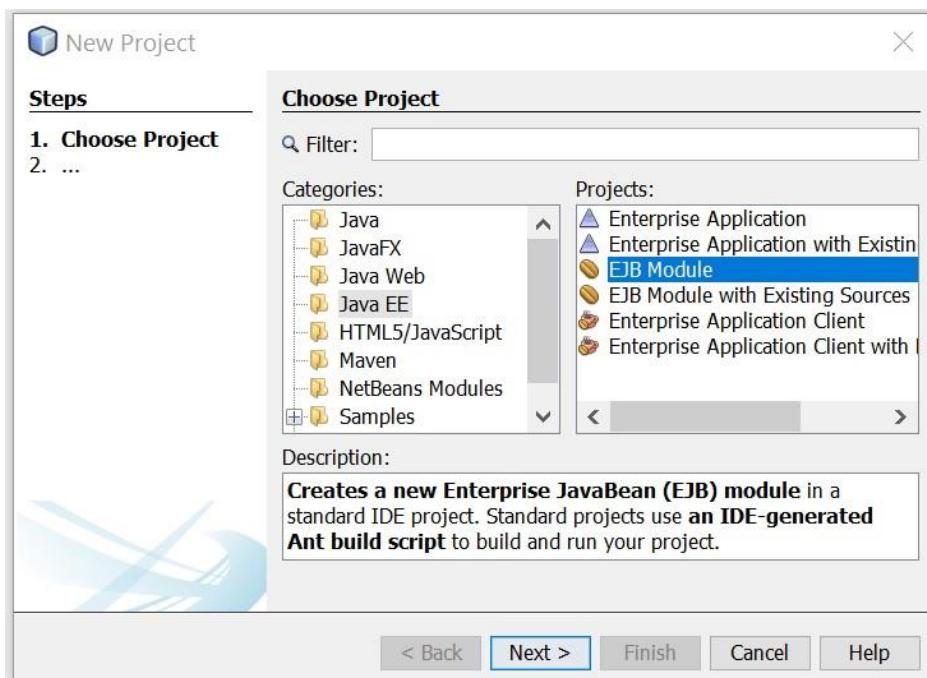
Location: C:\Users\User\Documents\NetBeansProjects

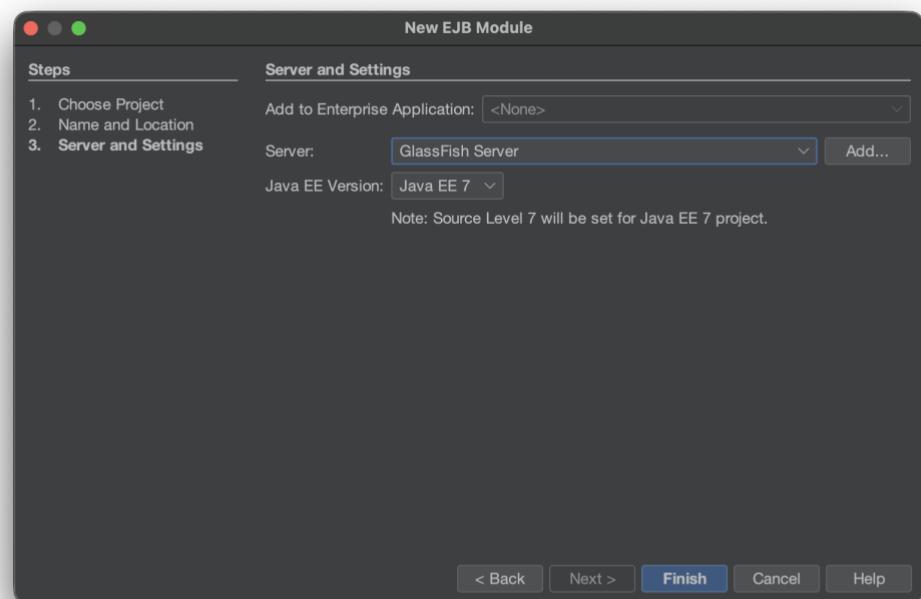
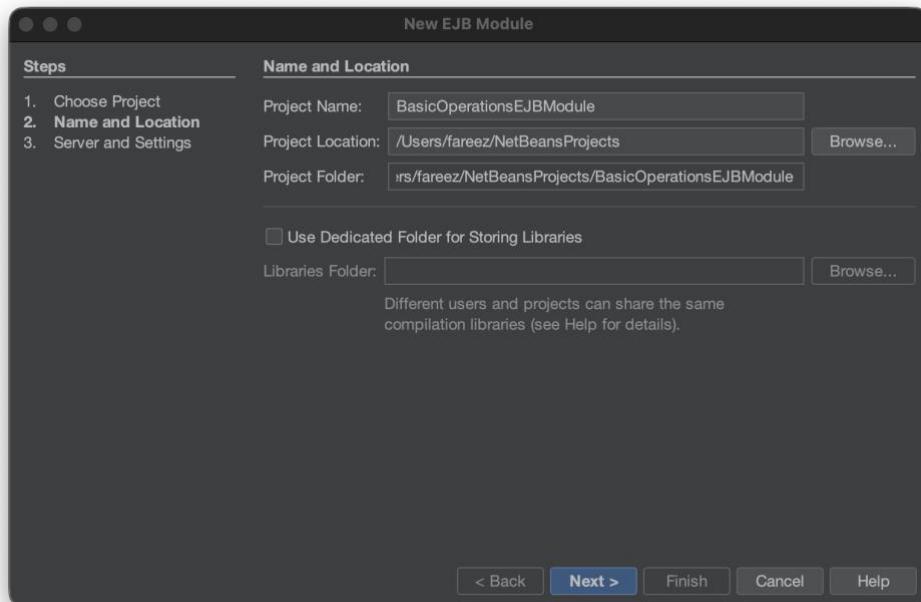
Folder: C:\Users\User\Documents\NetBeansProjects\ BasicOperationsEJBModule

Server : Personal glassfish

Java EE version: Java EE 7

Contextpath: / BasicOperationsEJBModule

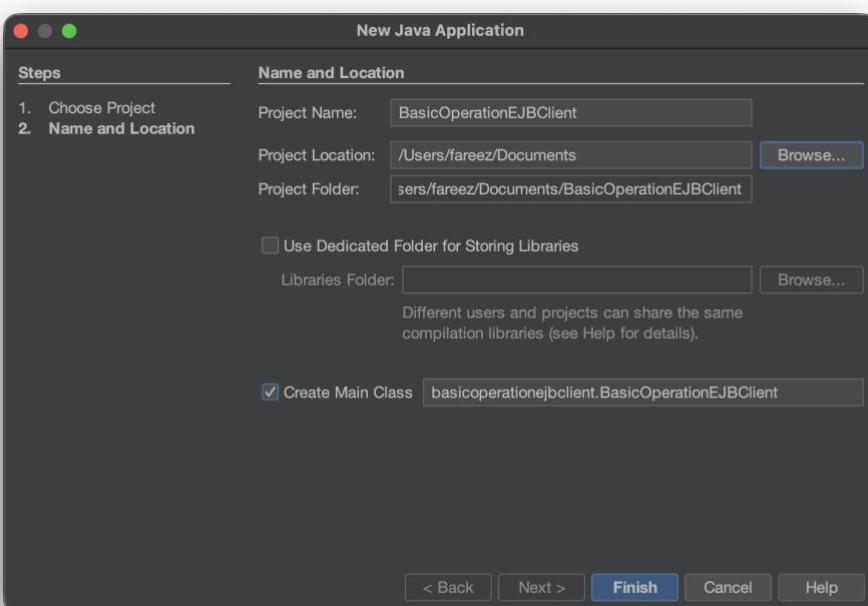
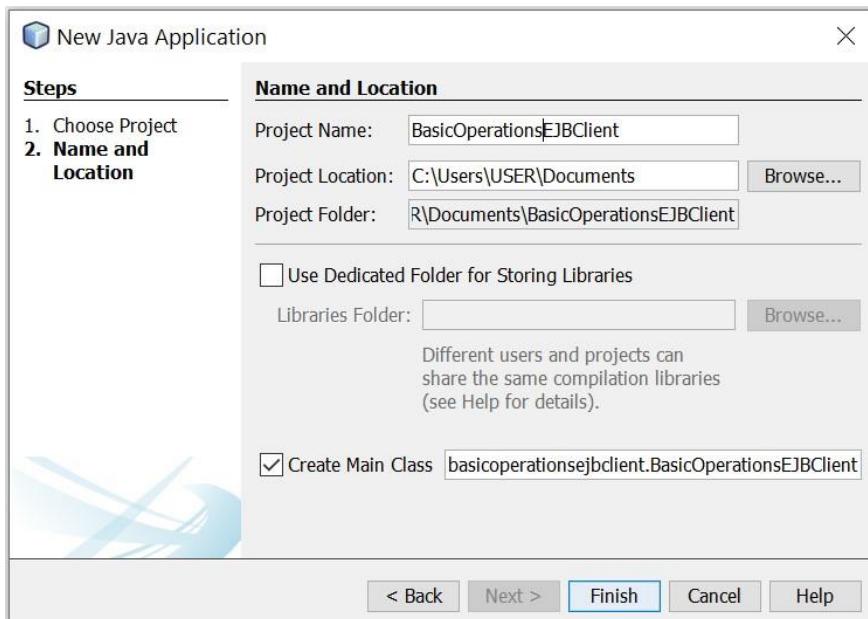




## Task 2: Create a new Application Class Project

In this section you will create an application class library project for EJB remote interface and entity class.

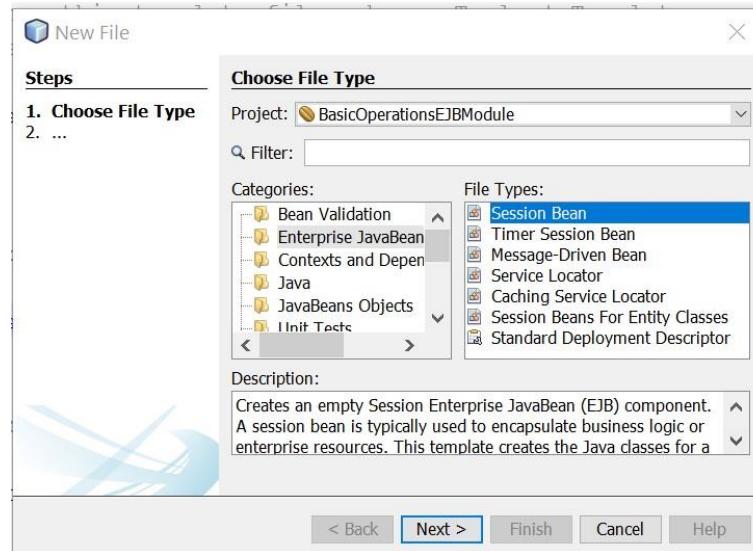
- Open NetBeans IDE, choose File > New Project. In the New Project wizard, expand the **Java category** and select **Java Application**. Then click Next.
- Specify the **Project Name** as BasicOperationsEJBClient and the **Project Location** in the appropriate text fields as shown in the figure and then click Finish.



**Task 3:** Create Session Bean

To create the stateless session bean, perform the following steps.

- Right-click the EJB module and choose New > Other to open the New File wizard.
- Select Session Bean in the Enterprise JavaBeans category. Click Next.



- In the next screen, specify **OperationsSessionBean** for the EJB Name and **com.ejb** for the Package name. Select **Stateless** and **Remote**. Then click Finish.

New Session Bean

**Steps**

1. Choose File Type  
2. Name and Location

**Name and Location**

EJB Name: OperationsSessionBean

Project: BasicOperationsEJBModule

Location: Source Packages

Package: com.ejb

Session Type:  
 Stateless  
 Stateful  
 Singleton

Create Interface:  
 Local  
 Remote in project: BasicOperationsEJBClient

< Back Next > **Finish** Cancel Help

New Session Bean

**Steps**

1. Choose File Type  
2. Name and Location

**Name and Location**

EJB Name: OperationsSessionBean

Project: BasicOperationsEJBModule

Location: Source Packages

Package: som.ejb

Session Type:  
 Stateless  
 Stateful  
 Singleton

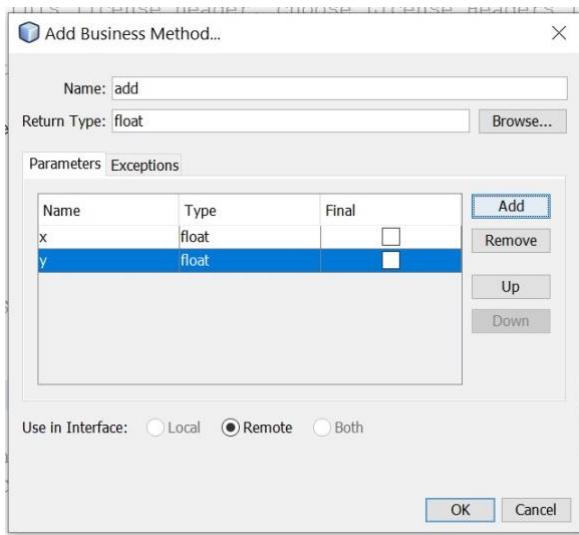
Create Interface:  
 Local  
 Remote in project: BasicOperationEJBClient

< Back Next > **Finish** Cancel Help

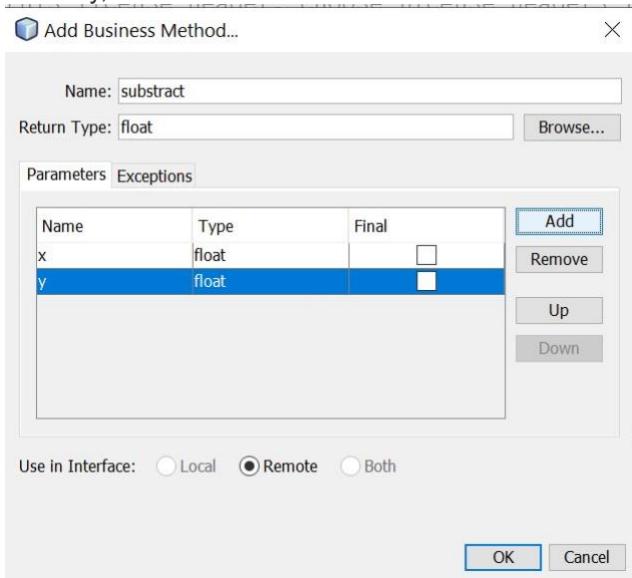
#### Task 4: Adding a business method

In this exercise you will create a simple business method in the session bean that returns a string.

- Right-click in the editor of **OperationsSessionBean** and choose Insert Code.
- Select Add Business Method.
- Type **add** in Method Name and float as return type and x, y as parameter name.



- Similarly, create **subtract** method as shown in below figure.

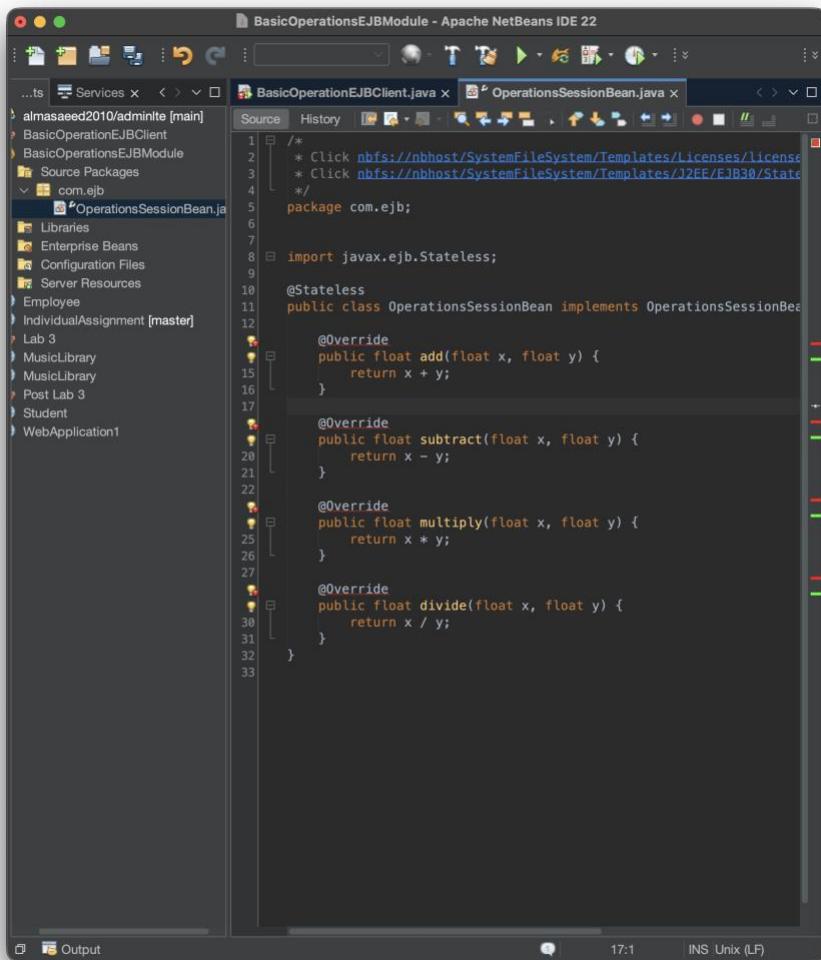


- Create **multiply** business method
- Create **divide** business method
- Modify the code for each business method as figure below.

```

6  package com.ejb;
7  import javax.ejb.Stateless;
8
9  @Stateless
10 public class OperationsSessionBean implements OperationsSessionBeanRemote {
11
12     @Override
13     public float add(float x, float y) {
14         return x + y;
15     }
16
17     @Override
18     public float subtract(float x, float y) {
19         return x - y;
20     }
21
22     @Override
23     public float multiply(float x, float y) {
24         return x * y;
25     }
26
27     @Override
28     public float divide(float x, float y) {
29         return x / y;
30     }
31
32 }
33

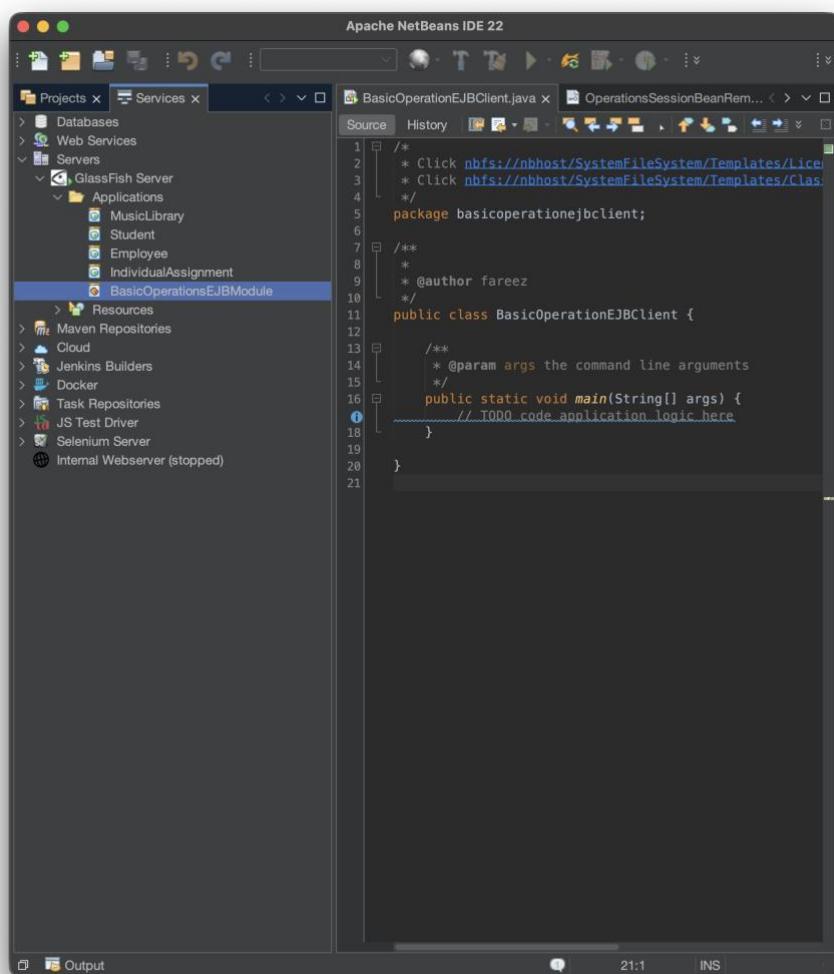
```



**Task 5: Deploy the EJB Module**

In this task, you can

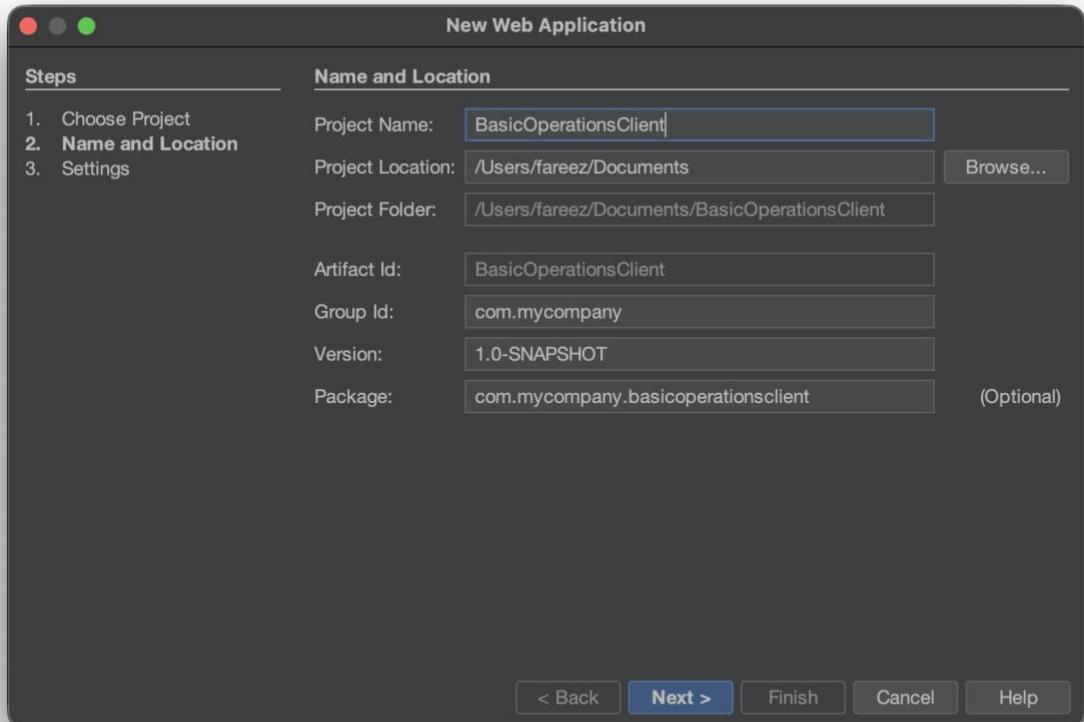
- a) Right-click the **BasicOperationsEJBModule** module and choose Deploy. When you click Deploy, the ejb module and deploys the JAR archive to the server.
- b) In the Services window, if you expand the Applications node of GlassFish Server you can see that was deployed.



c)

**Exercise 2: Create a new Web Application to test EJB**Task 1: Create a new **web application project** to test the EJB module

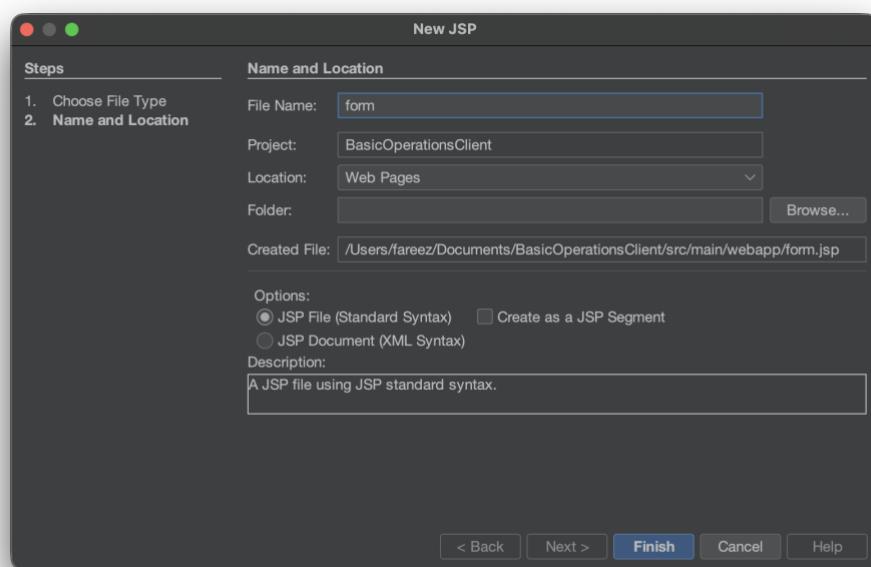
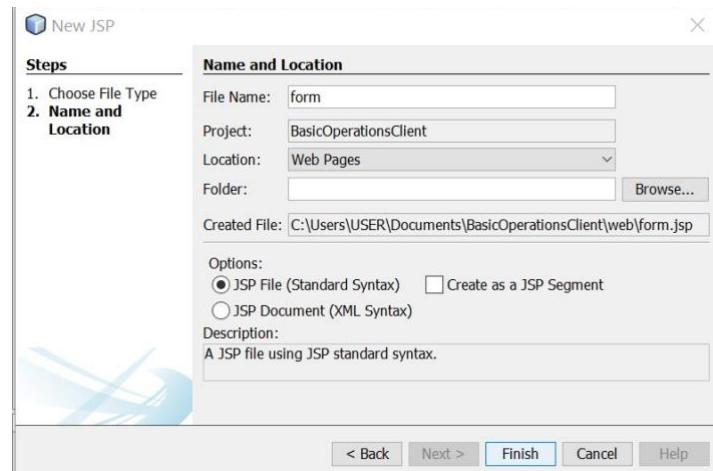
- Choose File > New Project. In the New Project wizard, expand the **Java Web** category and select **Web Application**. Then click Next.
- Specify the **Project Name** as **BasicOperationWebClient** and the Project Location in the appropriate text fields and then click Next.
- In the next window, add the **Java EE Server** as and select the **Java EE version** and click Finish.



Task 2: Create JSP Files to test EJB

In this exercise you will create a JSP to test user operations and obtain result.

- Right-click the Web module and choose File > New File wizard.
- In the New File wizard, expand the **Web category** and select **JSP**. Specify the **File name** as **form**. Then click Next.

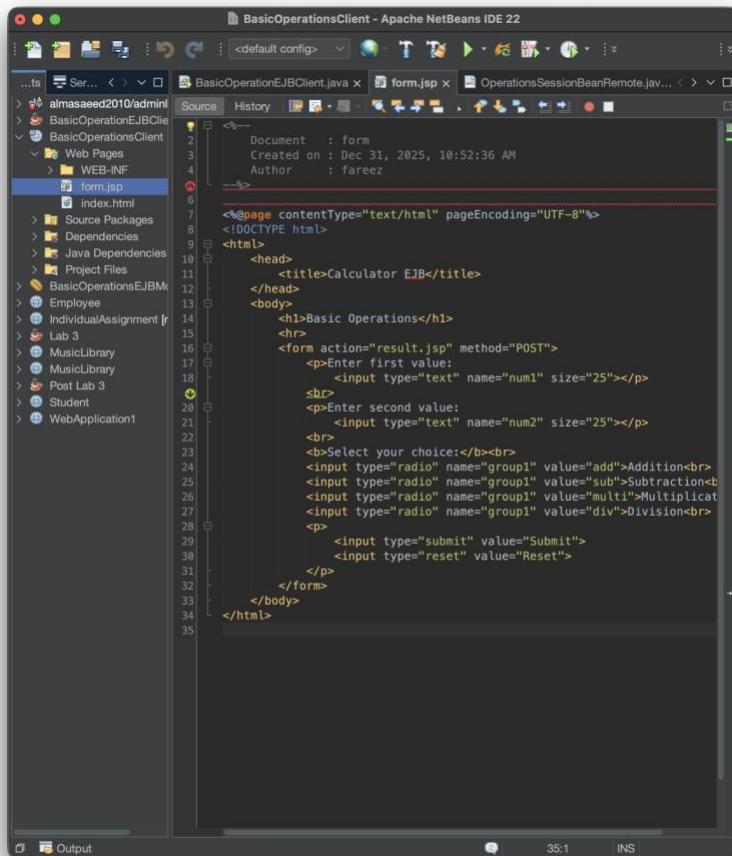


- c) Write the following code in the form.jsp.

```

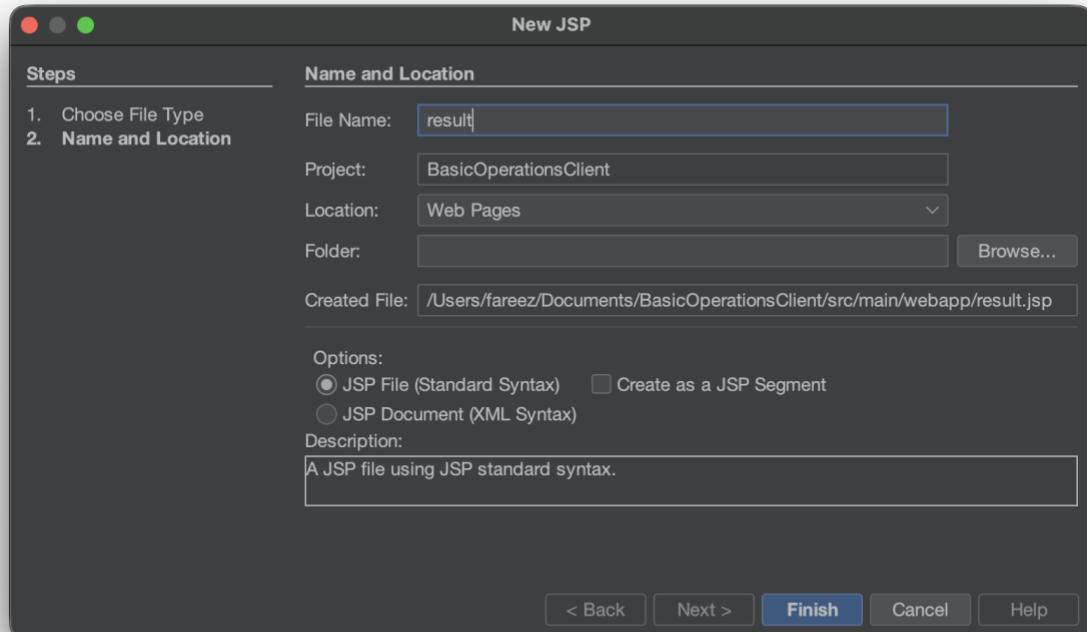
1 <html>
2   <head>
3     <title>Calculator EJB</title>
4   </head>
5   <h1>Basic Operations</h1>
6   <hr>
7   <form action="result.jsp" method="POST">
8     <p>Enter first value:
9       <input type="text" name="num1" size="25"></p>
10    <br>
11    <p>Enter second value:
12      <input type="text" name="num2" size="25"></p>
13    <br>
14    <b>Select your choice:</b><br>
15    <input type="radio" name="group1" value="add">Addition<br>
16    <input type="radio" name="group1" value="sub">Subtraction<br>
17    <input type="radio" name="group1" value="multi">Multiplication<br>
18    <input type="radio" name="group1" value="div">Division<br>
19    <p>
20      <input type="submit" value="Submit">
21      <input type="reset" value="Reset">
22    </p>
23  </form>
24 </body>
25 </html>

```



The result will be displayed in `result.jsp`. Create `result.jsp` as per below.

- Right-click the Web module and choose File > New File wizard.
- In the New File wizard, expand the **Web category** and select **JSP**. Specify the File name as `result.jsp`. Then click Next.



- Write the following code in the `result.jsp`.

```

1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <%@ page import="com.ejb.*; javax.naming.*" %>
3 <%!
4     private OperationsSessionBeanRemote ops = null;
5     float result = 0;
6
7     public void jspInit() {
8         try {
9             InitialContext ic = new InitialContext();
10            ops = (OperationsSessionBeanRemote)ic.lookup(OperationsSessionBeanRemote.class.getName());
11            System.out.println("Loaded Calculator Bean");
12        } catch (Exception ex) {
13            System.out.println("Error:" + ex.getMessage());
14        }
15    }
16
17    public void jspDestroy() {
18        ops = null;
19    }
20 %>
```

```
21  <%
22      try {
23          String s1 = request.getParameter("num1");
24          String s2 = request.getParameter("num2");
25          String s3 = request.getParameter("group1");
26
27          System.out.println(s3);
28
29          if (s1 != null && s2 != null) {
30              Float num1 = new Float(s1);
31              Float num2 = new Float(s2);
32
33              if (s3.equals("add")) {
34                  result = ops.add(num1.floatValue(), num2.floatValue());
35              } else if (s3.equals("sub")) {
36                  result = ops.subtract(num1.floatValue(), num2.floatValue());
37              } else if (s3.equals("multi")) {
38                  result = ops.multiply(num1.floatValue(), num2.floatValue());
39              } else {
40                  result = ops.divide(num1.floatValue(), num2.floatValue());
41              }
42
43      %>
44      <p>
45          <b>The result is:</b> <%= result%>
46      <p>
47          <%
48              // end of try
49          catch (Exception e) {
50              e.printStackTrace();
51              //result = "Not valid";
52          }
53
54      %>
```

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ page import="com.ejb.*, javax.naming.*" %>
<%!
    private OperationsSessionBeanRemote ops = null;
    float result = 0;

    public void jspInit() {
        InitialContext ic = new InitialContext();
        ops = (OperationsSessionBeanRemote)ic.lookup(OperationsSessionBeanRemote.class);
        System.out.println("Loaded Calculator Bean");
    } catch (Exception ex) {
        System.out.println("Error:" + ex.getMessage());
    }
}

public void jspDestroy() {
    ops = null;
}

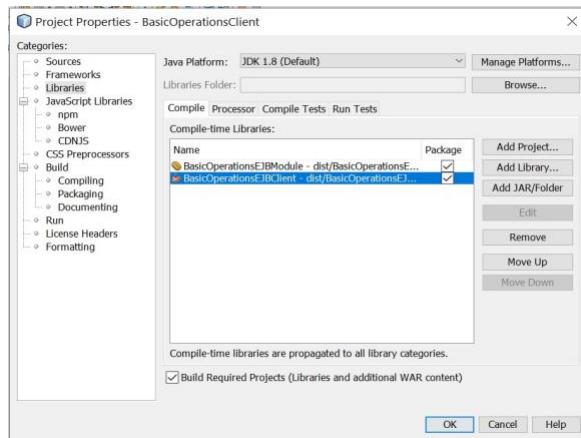
%>
<%
try {
    String s1 = request.getParameter("num1");
    String s2 = request.getParameter("num2");
    String s3 = request.getParameter("group1");

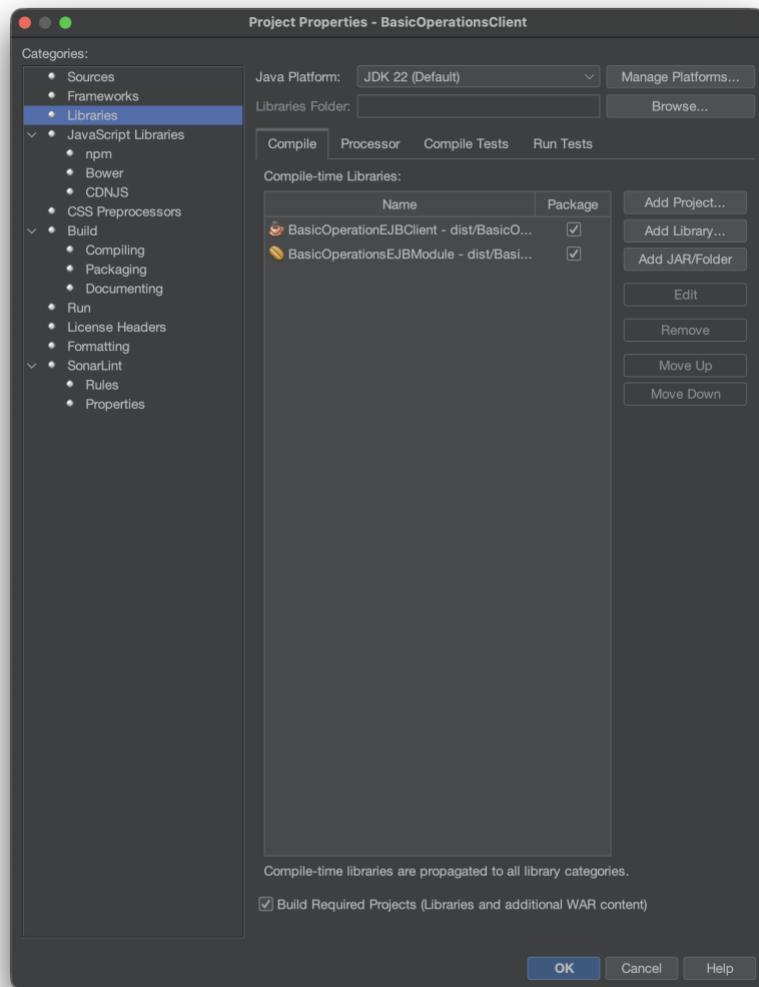
    System.out.println(s3);

    if (s1 != null && s2 != null) {
        Float num1 = new Float(s1);
        Float num2 = new Float(s2);

        if (s3.equals("add")) {
            result = ops.add(num1.floatValue(), num2.floatValue());
        } else if (s3.equals("sub")) {
            result = ops.subtract(num1.floatValue(), num2.floatValue());
        } else if (s3.equals("multi")) {
            result = ops.multiply(num1.floatValue(), num2.floatValue());
        } else {
            result = ops.divide(num1.floatValue(), num2.floatValue());
        }
    }
}
%>
<p> <b>The result is:</b> <%= result %>
<p>
<%
} // end of try
```

- d) Right click **BasicOperationsClient** project and select Properties
- e) In **BasicOperationsEJBModule** and **BasicOperationsEJBClient** the menu select **Libraries** click **Add Project** and add projects. Click OK.



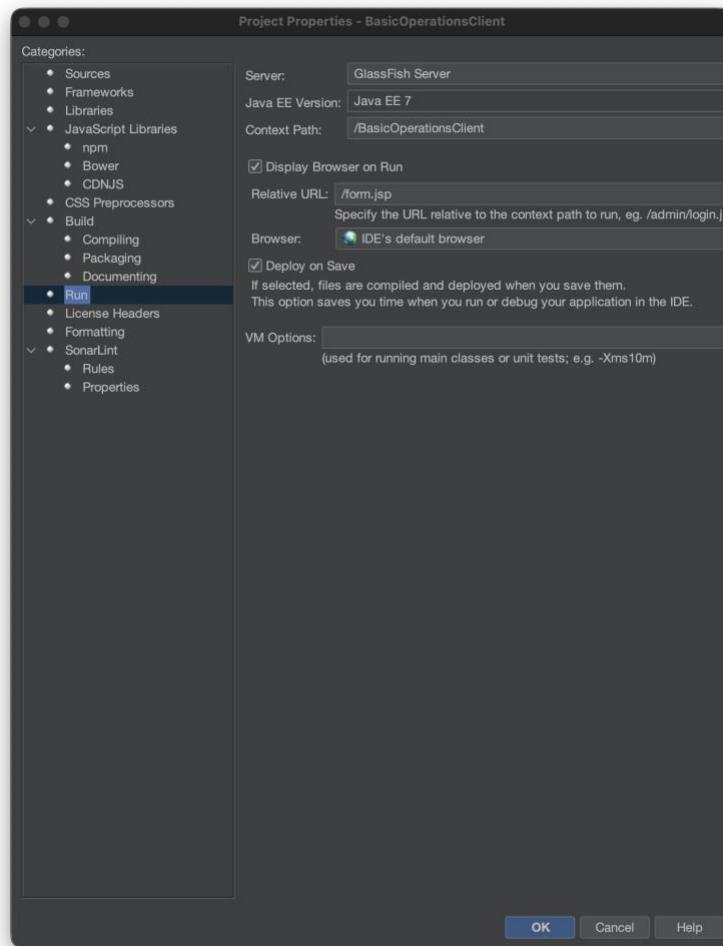


### Task 3: Run the Project

You can now run the project. When you run the project, you want the browser to open to the page with the **form.jsp**. You do this by specifying the URL in the Properties dialog box for the web application. The URL is relative to the context path for the application. After you enter the relative URL, you can build, deploy and run the application from the Projects window.

To set the **relative URL** and run the application, do the following:

- In the Projects window, right-click the **BasicOperationsWebClient** application node
- Select Properties in the pop-up menu.
- Select Run in the Categories pane.
- In the Relative URL textfield, type `/form.jsp`.
- Click OK.



**Basic Operations**

---

Enter first value:

Enter second value:

Select your choice:

Addition  
 Subtraction  
 Multiplication  
 Division

The result is: 200.0