

Week 2

# JSP: Scriptlet, Page Directive & Include Directive

Web Programming 2

JSP

---

Name:

NUR  
ARINA  
BINTI  
ABDUL  
MALEK

Matric #:

S65361

Semester;

4

Lab: 3

Demonstrator: SIR ARIZAL

---

Lecturers

PUSAT PENGAJIAN INFORMATIK DAN MATEMATIK  
GUNAAN (PPIMG), UNIVERSITI MALAYSIA TERENGGANU  
(UMT)

## Revision History

Revision Date	Previous Revision Date	Summary of Changes	Changes Marked
		First Issue	Mohamad Nor Hassan
		Second Issue	Dr Rabiei Mamat Dr Faizah Aplop Dr Fouad Ts Dr Rosmayati Mohemad Fakhrul Adli Mohd Zaki
21/02/2019		Addition of Revision History, Table of Contents, Formatting Cover Page	Fakhrul Adli Mohd Zaki

## Table of Contents

<b>Task 1: Passing Data from Main JSP's Page to Other JSP's Page .....</b>	<b>5</b>
<b>Task 2: Using Mathematics operations in JSP.....</b>	<b>10</b>
<b>Task 3: Populate an Array Values into HTML's Table .....</b>	<b>14</b>
<b>Task 4: Perform Calculation of Car Loan .....</b>	<b>16</b>
<b>Task 5: Using JSP Page Directive to Call Java API.....</b>	<b>19</b>
<b>Task 6: Use JSP Include directive for JSP Page.....</b>	<b>25</b>

**Arahan:**

Manual makmal ini adalah untuk kegunaan pelajar-pelajar Pusat Pengajian Informatik dan Matematik Gunaan (PPIMG), Universiti Malaysia Terengganu (UMT) sahaja. Tidak dibenarkan mencetak dan mengedar manual ini tanpa kebenaran rasmi daripada penulis.

Sila ikuti langkah demi langkah sebagaimana yang dinyatakan di dalam manual. Tandakan (✓) setiap langkah yang telah selesai dibuat dan tulis kesimpulan bagi setiap aktiviti yang telah selesai dijalankan.

**Instruction:**

*This laboratory manual is for use by the students of the School of Informatics and Applied Mathematics (PPIMG), Universiti Malaysia Terengganu (UMT) only. It is not permissible to print and distribute this manual without the official authorisation of the author.*

*Please follow step by step as described in the manual. Tick (✓) each step completed and write the conclusions for each completed activity.*

## Task 1: Passing Data from Main JSP's Page to Other JSP's Page

- Objective** : To demonstrate the use of `request.getParameter("fieldName")` for passing input from one JSP's page to another JSP's page.
- Problem** : i. Create a page `memberRegister.jsp`.
- Description** ii. Page `memberRegister.jsp` consists of two (2) inputs;
- IC No (Must be in pre-formatted XXXXXXXXXX)
  - Name
3. In `memberRegister.jsp`, include two (2) buttons;  
*Submit* and *Cancel* button.
4. Create a page `memberProcessing.jsp`.
5. When user click *Submit* button, process the request and display the input key-in in `memberProcessing.jsp` page.
- Estimated time** : 30 minutes

1. Create new Project namely *Lab2*.
2. To create a JSP's page, right click *Lab2* -> *New* -> *JSP*.



3. Key-in File Name: *memberRegister*.

**New JSP**

**Steps**

1. Choose File Type
2. Name and Location

**Name and Location**

File Name:

Project:

Location:

Folder:

Browse...

Created File:

**Options:**

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

**Description:**

< Back   Next >   **Finish**   Cancel   Help

4. Click *Finish* button.

5. Source code for *memberRegister.jsp* will appear.

6. Write a HTML's markup to produce HTML's form

```
<body>
  <h1>Passing data from main JSP's page to other JSP's page </h1>
  <form id="memberFrm" action="memberProcessing.jsp" method="post" onsubmit="return checkICNo()">
    <fieldset>
      <legend>Member Registration</legend>
      <label for="invoiceno">Ic No *</label>
      <input type="text" id="icno" name="my_icno" size="15" placeholder="E.g. 921012101245"><br/>
      <label for="name">Name</label>
      <input type="text" id="name" name="my_name" size="45" placeholder="Key-in your name"><br/>
      <p><input type="submit" id="btnSubmit" value="Submit"/>
        <input type="reset" id="btnCancel" value="Cancel"/>
      </p>
    </fieldset>
  </form>
</body>
```

7. Save and compile *memberRegister.jsp* file.

8. Run the *memberRegister.jsp* file and you should get the interface as below:



Lab

localhost:8084/Lab2/memberRegister.jsp

Apps Zimbra Sign In Lentera OneDrive HTML5: The Missing MyNemo ezHASiL Free Code Download

## Passing data from main JSP's page to other JSP's page

Member Registration

Id No \*

Name

©2016-Mohamad Nor

9. Repeat step 1 and step 2.

10. Key-in File Name: *memberProcessing*.

11. Click *Finish* button.

12. Source code for *memberProcessing.jsp* will appear.

13. Write a HTML code.

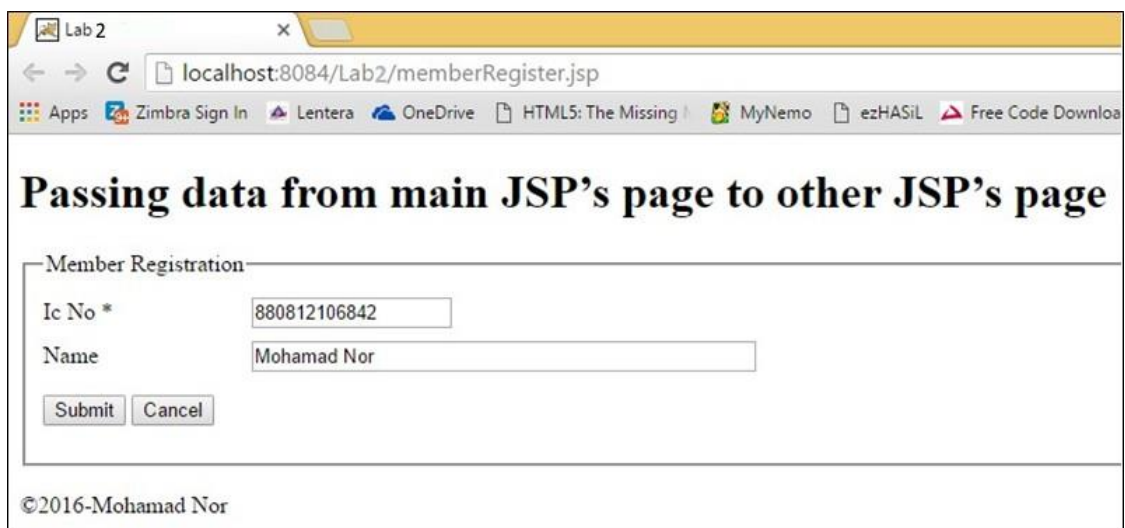
```
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>Lab 6 - Task 6</title>
13 </head>
14 <body>
15 <h1>Passing data from main JSP's page to other JSP's page </h1>
16
17 </body>
18 </html>
```

14. Add additional HTML's tag and Java Scriptlet to retrieve the value from main's form.

```
16 <fieldset>
17 <%
18     //Define variables...
19     String myIC = null;
20     String myName = null;
21
22     //Use request.getParameter() method to retrieve data from main's form...
23     myIC = request.getParameter("my_icno");
24     myName = request.getParameter("my_name");
25 >%
26
27 <!-- Display the output... -->
28 <p>Thank you for registering in this event..!</p>
29 <p>This is your details;</p>
30 <p>IC No : <%=myIC%></p>
31 <p>Name : <%=myName%></p>
32 </fieldset>
```

15. Compile *memberProcessing.jsp* file.

16. Run the *memberRegister.jsp* file and fill-up the input.



Lab 2

localhost:8084/Lab2/memberRegister.jsp

Apps Zimbra Sign In Lentera OneDrive HTML5: The Missing MyNemo ezHASiL Free Code Downloa

## Passing data from main JSP's page to other JSP's page

Member Registration

Ic No \*

Name

©2016-Mohamad Nor

17. Click *Submit* button to send the request.



18. These inputs will be sent to *memberProcessing.jsp* page and produce the following page.



## Mywork

### memberRegister.jsp

```
1 <!--
2 Document : memberRegister
3 Created on : 17 Apr 2024, 2:39:06 pm
4 Author : ryna
5 --%>
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15 <h1>Passing data from main JSP's page</h1>
16 <form id="memberFrm" action="memberProcessing.jsp" method="post" onsubmit="return checkICNo()">
17 <fieldset>
18 <legend>Member Registration</legend>
19 <label for="invoiceno">Ic No *</label>
20 <input type="text" id="icno" name="my_icno" size="15" placeholder="E.g. 921012101245"><br/>
21
22 <label for="name">Name</label>
23 <input type="text" id="name" name="my_name" size="45" placeholder="Key-in your name"><br/>
24
25 <p><input type="submit" id="btnSubmit" value="Submit"/>
26 <input type="reset" id="btnCancel" value="Cancel"/>
27 </p>
28 </form>
29 </body>
30 </html>
31
```



### Passing data from main JSP's page

Member Registration

Ic No \* 880812106842

Name Mohamad Nor

Submit

Cancel

## memberProcessing.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Lab 3 - Task 1</title>
</head>
<body>
<h1>Passing data from main JSP's page to other JSP's page</h1>

<fieldset>
<%
// Define variables
String myIC = null;
String myName = null;

// Use request.getParameter() method to retrieve data from main's form
myIC = request.getParameter("my_icno");
myName = request.getParameter("my_name");

%>

<!-- Display the output -->
<p>Thank you for registering in this event..!</p>
<p>This is your details;</p>
<p>IC No : <%= myIC %></p>
<p>Name : <%= myName %></p>

</fieldset>
</body>
</html>
```

## Passing data from main JSP's page to other JSP's page

Thank you for registering in this event..!

This is your details;

IC No : 880812106842

Name : Mohamad Nor

## Reflection

1. How do you want to submit specific information from one form to next form?  
= using session Attributes

2. What happened if the field name you specify in  
`request.getParameter("field_name")` in second page is different from the field  
name you defined in first page?

= it will return nothing and error

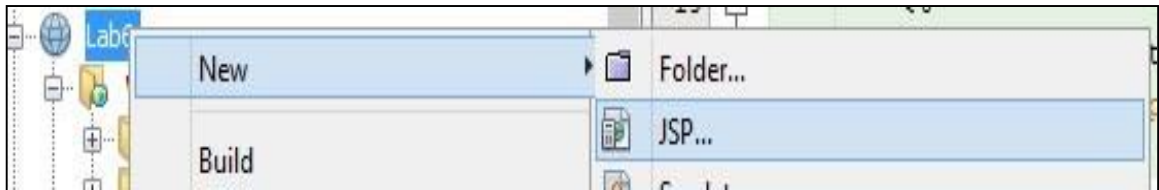
## Task 2: Using Mathematics Operations in JSP

**Objective** : To demonstrate the use of *request.getParameter* (“*Mathematics operations*”) in JSP’s page.

**Problem Description** : i. Create a page *Calculator.jsp* consists of interface represent basic calculator.  
ii. When user key-in inputs, process the request and display the results direct in JSP page.

**Estimated time** : 30 minutes

1. Go to Project *Lab2*.
2. To create a JSP’s page, right click *Lab2* -> *New* -> *JSP*.



3. Key-in File Name: *Calculator*.

**Steps**

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

**Name and Location**

File Name:

Project:

Location:

Folder:

Created File:

Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

< Back   Next >   **Finish**   Cancel   Help

4. Click *Finish* button.

5. Source code for *Calculator.jsp* will appear.

6. Write a HTML's markup to produce HTML's form

```
<body bgcolor= "#a00FFF" text= "gold">

<center>

<h2>Basic calculator program in jsp</h2>
<form method ="get" name ="f1">
<input type ="text" size ="20" name ="operand1" value = "" />

<select name = op size = 1>
<option value = "0" >+</option>
<option value = "1" >-</option>
<option value = "2" >*</option>
<option value = "3" >/</option>
<option value = "4" >%</option>
</select>

<input type ="text" size="20" name ="operand2" value = ""/>
<p><br><br><br><br>
<input type = submit value = Calculate />

</form>

</body>
```

7. Save and compile *Calculator.jsp* file.

8. Run the *Calculator.jsp* file and you should get the interface as below:

### Basic calculator program in jsp

+

+

-

\*

/

%

Result = 0

9. Add additional HTML's tag and Java scriptlet to retrieve the value from users.

```
<%  
String num1 = "0", num2 = "0";  
int result = 0;  
String op = "+";  
  
char opchar = op.charAt(0);  
if (request.getParameter("op") != null) {  
    op = request.getParameter("op");  
    opchar = op.charAt(0);  
  
    num1 = request.getParameter("operand1");  
    num2 = request.getParameter("operand2");  
  
    switch(opchar) {  
        case '0': result = Integer.parseInt(num1) + Integer.parseInt(num2);  
        break;  
        case '1': result = Integer.parseInt(num1) - Integer.parseInt(num2);  
        break;  
        case '2': result = Integer.parseInt(num1) * Integer.parseInt(num2);  
        break;  
        case '3': result = Integer.parseInt(num1) / Integer.parseInt(num2);  
        break;  
        case '4': result = Integer.parseInt(num1) % Integer.parseInt(num2);  
        break;  
    }  
}  
%>  
  
Result = <%= result + " ">
```

9. Further, add additional Java Scriplet to HTML's tag as below.

```
<body bgcolor= "#a00FFF" text= "gold">
<center>

<h2>Basic calculator program in jsp</h2>
<form method ="get" name ="f1">
<input type ="text" size ="20" name ="operand1" value = <%= num1 %> />

<select name = op size = 1>
<option value = "0" >+</option>
<option value = "1" >-</option>
<option value = "2" >*</option>
<option value = "3" >/</option>
<option value = "4" >%</option>
</select>

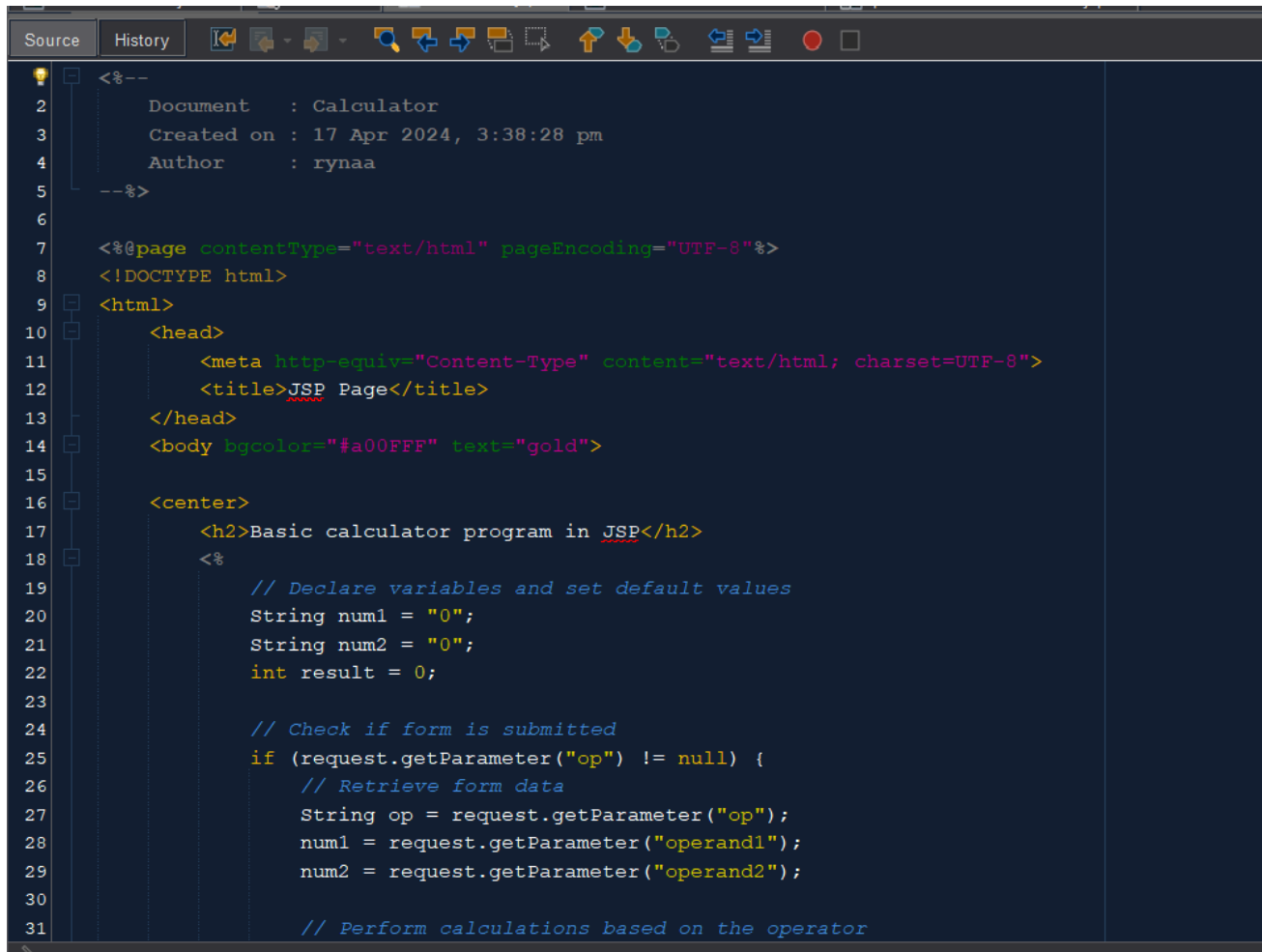
<input type ="text" size="20" name ="operand2" value = <%= num2 %> />
<p>
<input type = submit value = Calculate />

Result = <%= result + " " %>
</form>
</body>
```

10. Compile *Calculator.jsp* file.

11. Run the *Calculator.jsp* file and test the clculator.

## MY WORK



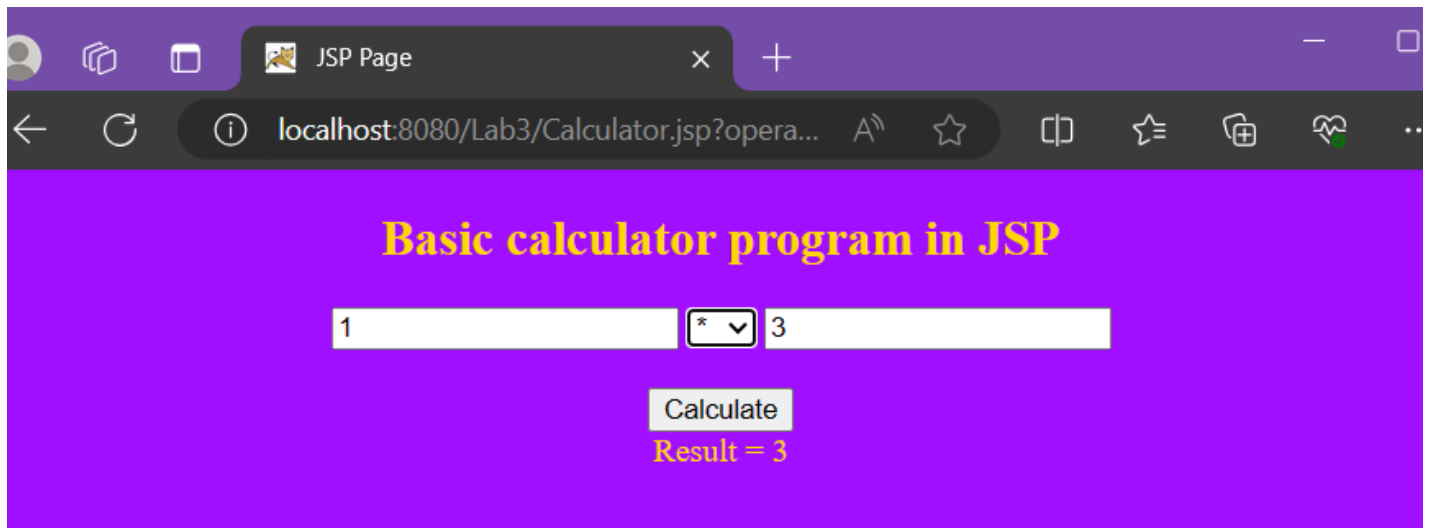
```
<%--
2      Document      : Calculator
3      Created on    : 17 Apr 2024, 3:38:28 pm
4      Author       : rynaa
5  --%>
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <!DOCTYPE html>
9  <html>
10     <head>
11         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12         <title>JSP Page</title>
13     </head>
14     <body bgcolor="#a00FFF" text="gold">
15
16     <center>
17         <h2>Basic calculator program in JSP</h2>
18         <%
19             // Declare variables and set default values
20             String num1 = "0";
21             String num2 = "0";
22             int result = 0;
23
24             // Check if form is submitted
25             if (request.getParameter("op") != null) {
26                 // Retrieve form data
27                 String op = request.getParameter("op");
28                 num1 = request.getParameter("operand1");
29                 num2 = request.getParameter("operand2");
30
31                 // Perform calculations based on the operator
```



```

31 // Perform calculations based on the operator
32 switch (op) {
33     case "0":
34         result = Integer.parseInt(num1) + Integer.parseInt(num2);
35         break;
36     case "1":
37         result = Integer.parseInt(num1) - Integer.parseInt(num2);
38         break;
39     case "2":
40         result = Integer.parseInt(num1) * Integer.parseInt(num2);
41         break;
42     case "3":
43         // Check for division by zero
44         int divisor = Integer.parseInt(num2);
45         if (divisor != 0) {
46             result = Integer.parseInt(num1) / divisor;
47         } else {
48             out.println("Error: Division by zero!");
49         }
50         break;
51     case "4":
52         result = Integer.parseInt(num1) % Integer.parseInt(num2);
53         break;
54     default:
55         out.println("Error: Invalid operator!");
56         break;
57 }
58 }
59 %>
60
61 <form method="get" name="f1">
62
63     }
64     %>
65
66     <form method="get" name="f1">
67         <input type="text" size="20" name="operand1" value="<%= num1 %>" />
68
69         <select name="op" size="1">
70             <option value="0">+</option>
71             <option value="1">-</option>
72             <option value="2">*</option>
73             <option value="3">/</option>
74             <option value="4">%</option>
75         </select>
76
77         <input type="text" size="20" name="operand2" value="<%= num2 %>" />
78         <br><br>
79         <input type="submit" value="Calculate" />
80     </form>
81
82     <!-- Display result if calculated -->
83     <% if (request.getParameter("op") != null) { %>
84         Result = <%= result %>
85     <% } %>
86
87 </center>
88 </body>
89 </html>

```



### Reflection

1. How do you want to submit specific information from one form to next form?

= using sessions Attributes

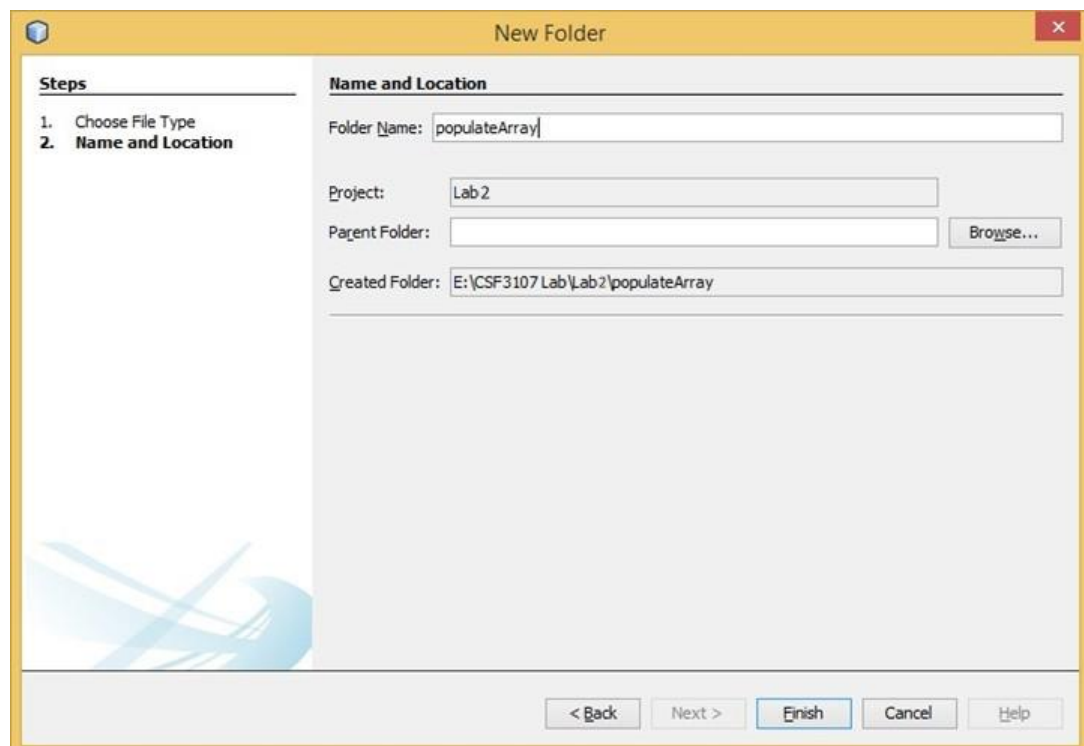
2. What happened if the field name you specify in `request.getParameter("field_name")` in second page is different from the field name you defined in first page?

= it will return nothing and error

## Task 3: Populate an Array Values into HTML's Table

- Objective** : Read Java array and populate it into HTML's table.
- Problem** : i. Create 2D array that store sales data.
- Description** ii. Then, read an array and populate into HTML's table.
- Estimated time** : 50 minutes

1. Go to Project *Lab2*.
2. To create a JSP's page, right click *Lab2* -> *New* -> *JSP*.
3. Key-in File Name: *populateArray*.



4. Click *Finish* button.

5. Prepare standard HTML's a markup for page *populateArray.jsp*.
6. Write a Java Scriptlet and store the following information into an array;

	Jan	Feb	Mac
Salesman 1	2500	2100	2200
Salesman 2	2000	1900	2400
Salesman 3	1800	2200	2450

7. Read the array and populate its value into HTML's table.
8. Save and compile *populateArray.jsp* file.
9. Run the *populateArray.jsp* file and sample of output is shown below:



Salesman	Jan	Feb	Mac
Salesman 1	2500	2100	2200
Salesman 2	2000	1900	2400
Salesman 3	1800	2200	2450
©2016-Mohamad Nor			

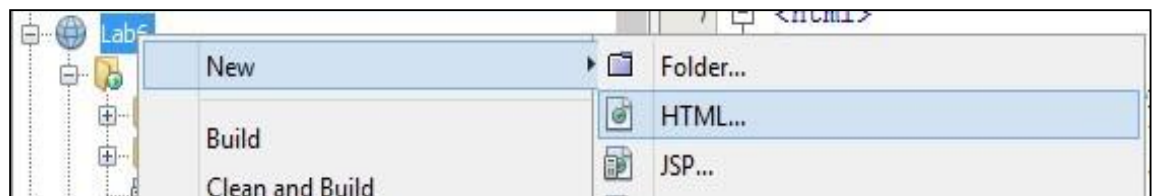
## Reflection

1. Write a sample syntax to declare 2D Java array.
2. Define a sequence of steps on how you accomplish Task 7.
3. What is the difference between HTML's page and JSP's page

## Task 4: Perform Calculation of Car Loan

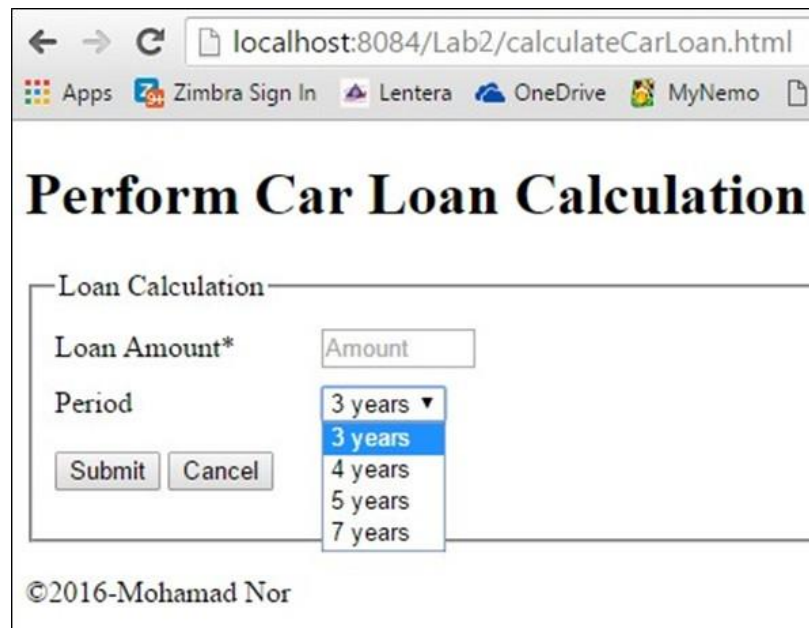
<b>Objective</b>	: Passing input to next page for further processing.
<b>Problem Description</b>	: i. Create simple interface in HTML that consists of Loan Amount and Loan Period. ( Loan Period < 5 years, interest is 2.8% per year, and > 5 years interest is 4.5% per year).  ii. Submit the form and perform calculation based on user input and, finally, display the result.
<b>Estimated time</b>	: 50 minutes

1. Go to Project Lab2.
2. To create a HTML's page, right click Lab2 -> New -> JSP



3. Key-in File Name: *calculateCarLoan*
4. Create a standard HTML's markup for form.
5. In your form, create two (2) fields; *Loan Amount* and *Loan Period*.
6. Save *calculateCarLoan.html* and run the file.

7. You will get the following output.



← → ↻ localhost:8084/Lab2/calculateCarLoan.html

Apps Zimbra Sign In Lentera OneDrive MyNemo

## Perform Car Loan Calculation

Loan Calculation

Loan Amount\*

Period

©2016-Mohamad Nor

8. Create JSP's file and rename the file as *processCalculateCarLoan*.

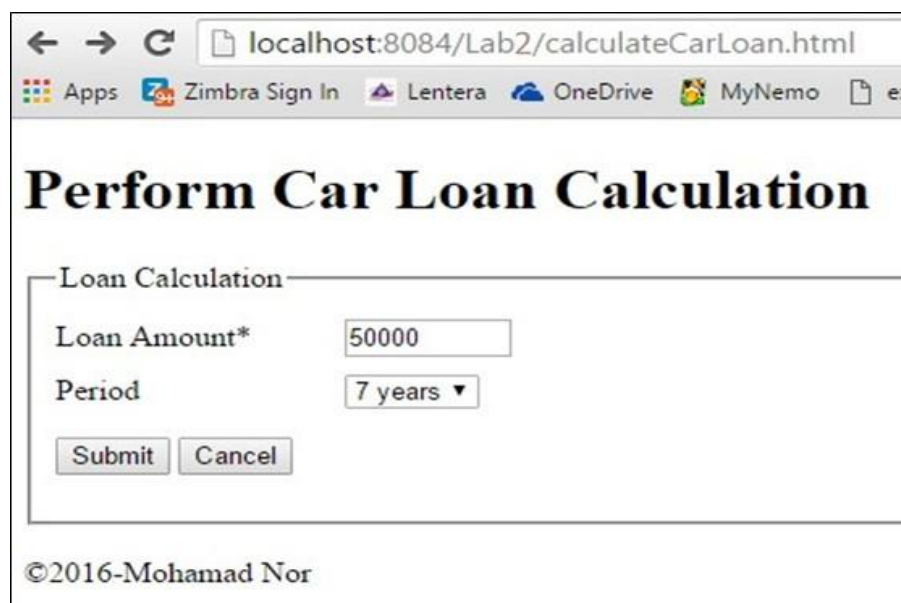
9. Construct the logic for calculating car loan and display the result.

10. Save and compile *processCalculateCarLoan.jsp*.

11. Run *calculateCarLoan.html* file and fill-up the input.

12. Then, submit your result.

13. You should get the following output;



← → ↻ localhost:8084/Lab2/calculateCarLoan.html

Apps Zimbra Sign In Lentera OneDrive MyNemo

## Perform Car Loan Calculation

Loan Calculation

Loan Amount\*

Period

©2016-Mohamad Nor

localhost:8084/Lab2/processCalculateCarLoan.jsp

Apps Zimbra Sign In Lentera OneDrive MyNemo ezHASiL M

## Perform Car Loan Calculation

### Details of car loan:

Loan Request : 50000

Period of payment : 7

Total Loan (+ interest) : 65750.00

©2016-Mohamad Nor

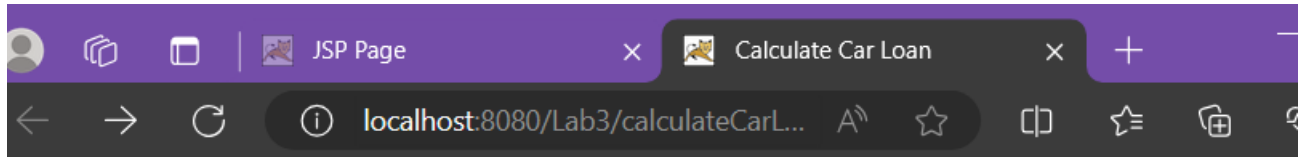
## MY WORK

### calculateCarLoan.jsp

```

Source History
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Calculate Car Loan</title>
5 </head>
6 <body>
7 <h1>Perform Car Loan Calculation</h1>
8 <form action="processCalculateCarLoan.jsp" method="post">
9 <label for="loanAmount">Loan Amount*:</label>
10 <input type="number" id="loanAmount" name="loanAmount" required><br><br>
11
12 <label for="Period">Period:</label>
13 <select id="Period" name="Period">
14 <option value="3">3 years</option>
15 <option value="4">4 years</option>
16 <option value="5">5 years</option>
17 <option value="7">7 years</option>
18 </select><br><br>
19
20 <input type="submit" value="Submit">
21 <button type="button" onclick="cancelForm()">Cancel</button>
22 </form>
23
24 <script>
25 function cancelForm() {
26 alert("Form canceled!");
27 }
28 </script>
29 </body>
30 </html>
31
32
33
34
35
36
37

```



## Perform Car Loan Calculation

Loan Amount\*:

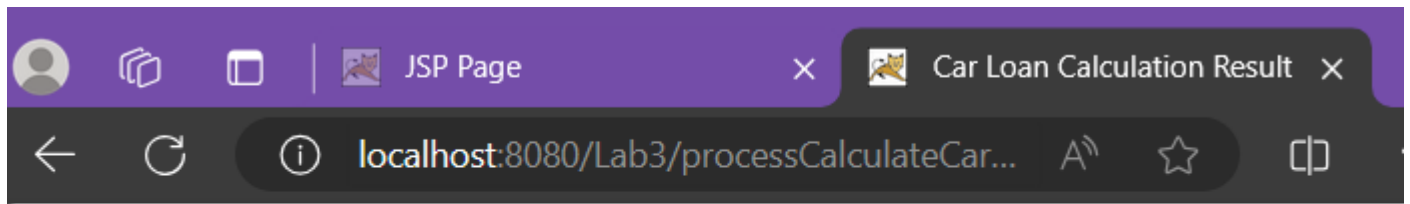
Period:



## processCalculateCarLoan.jsp

```
7 <%@ page contentType="text/html; charset=UTF-8" %>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <title>Car Loan Calculation Result</title>
12 </head>
13 <body>
14 <h2>Car Loan Calculation Result</h2>
15
16 <!-- Get parameters from the form -->
17 <%
18     String loanAmountStr = request.getParameter("loanAmount");
19     String loanPeriodStr = request.getParameter("Period"); // Corrected variable name
20
21     // Validate inputs
22     if (loanAmountStr != null && loanPeriodStr != null && !loanAmountStr.isEmpty() && !loanPeriodStr.isEmpty()) {
23         try {
24             double loanAmount = Double.parseDouble(loanAmountStr);
25             int loanPeriod = Integer.parseInt(loanPeriodStr);
26
27             if (loanAmount <= 0 || loanPeriod <= 0) {
28
29                 <p>Error: Loan amount and period must be positive numbers.</p>
30             } else {
31                 // Calculate monthly payment
32                 double interestRate = 0.05; // 5% interest rate (adjust as needed)
33                 double monthlyInterestRate = interestRate / 12;
34                 int numberOfPayments = loanPeriod * 12;
35                 double monthlyPayment = (loanAmount * monthlyInterestRate) / (1 - Math.pow(1 + monthlyInterestRate, -numberOfPayments));
36
37
```

```
37
38         // Calculate total payment (loan + interest)
39         double totalPayment = monthlyPayment * numberOfPayments;
40
41         // Display result
42         <p>Loan Request: <%= loanAmount %> </p>
43         <p>Period of Payment: <%= loanPeriod %> </p>
44         <p>Total Loan (+ Interest): <%= String.format("%.2f", totalPayment) %> </p>
45     }
46 <%
47     }
48 } catch (NumberFormatException e) {
49     // Handle invalid input format
50
51     <p>Error: Invalid input format. Please enter valid numbers.</p>
52 }
53
54 } else {
55     // Handle missing input
56
57     <p>Error: Missing input. Please fill in all fields.</p>
58 }
59 <%
60 }
61 </body>
62 </html>
```



## Car Loan Calculation Result

Loan Request: 50000.0

Period of Payment: 7

Total Loan (+ Interest): 59362.42

### Reflection

1. How you want to retrieve data from previous page?  
= use code `request.getParameter("parameterName")`

2. Where the construction of logic occur for calculating Total Loan ( + interest) ?  
= in the server-side processing jsp

## Task 5: Using JSP Page Directive to Call Java API

**Objective** : Use JSP page directive elements to call certain Java API.

**Problem** : Using Java *ArrayList* object to store data and retrieve it via JSP page.

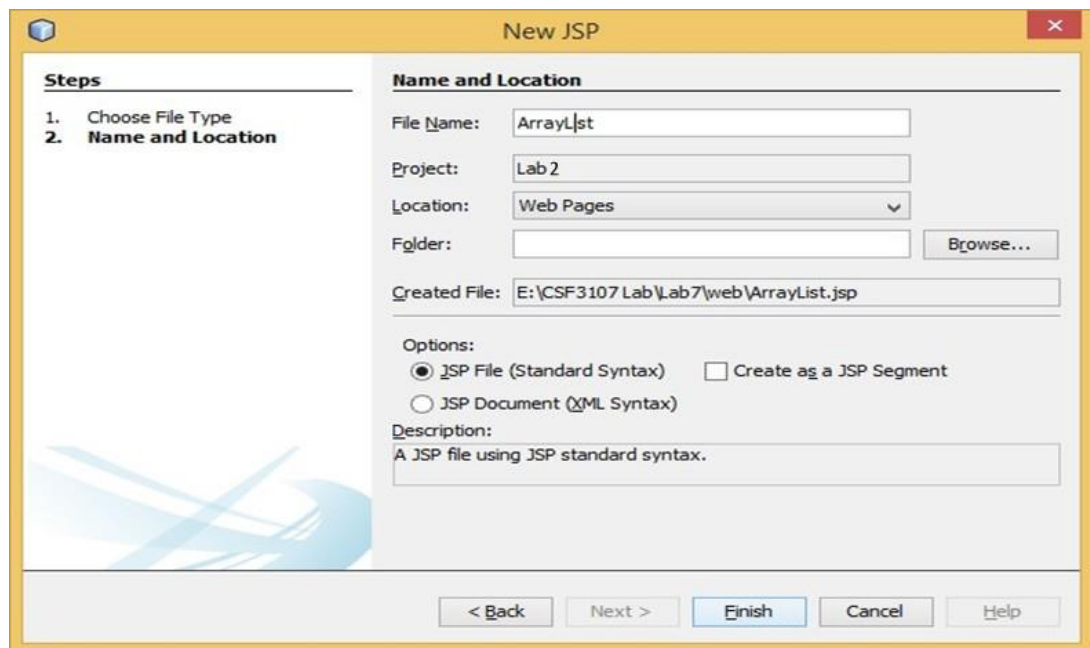
**Description**

**Estimated time** : 20 minutes

1. Create a new JSP's file.



2. Type file name as *ArrayList*.



2. Click *Finish* button.

4. Type title as *Use Java ArrayList*.

5. Type header1 as *Use JSP Page Directive*

```
1 <%--
2     Document    : ArrayList
3     Created on  : 10-Apr-2016, 09:24:46
4     Author     : Mohamad Nor Hassan
5 --%>
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12     <title>Use Java ArrayList</title>
13 </head>
14 <body>
15     <h1>Use JSP Page Directive</h1>
16 </body>
17 <br/>
18 <footer>&copy;2016-Mohamad Nor</footer>
19 </html>
```

6. In order to use Java *ArrayList*'s object, we need to use JSP Page Directive and import the related API.

```
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@page import="java.util.ArrayList"%>
```

7. In order to use Java syntax, create a Java Scriptlet notation.

```
15 <body>
16     <h1>Use JSP Page Directive</h1>
17     <%
18
19     %>
20 </body>
```

8. Create an object *ArrayList* to store a list of student name.

```
17 <%
18     //Create ArrayList object ...
19     ArrayList<String> studentList = new ArrayList<String>();
20 %>
```

9. Add the following name to ArrayList's object.

- ✓ Mohamad Azam
- ✓ Peter Chong
- ✓ Rahimah Mansor
- ✓ Sri Devi
- ✓ Ng Hue Ween
- ✓ S. Nagarajan

```
21 //Store student name..  
22 studentList.add(0, "Mohamad Azam");  
23 studentList.add(1, "Peter Chong");  
24 studentList.add(2, "Rahimah Mansor");  
25 studentList.add(3, "Sri Devi");  
26 studentList.add(4, "Ng Hue Ween");  
27 studentList.add(5, "S. Nagarajan");
```

10. Display the number of records for an ArrayList's object.

```
28  
29 //Display the number of records..  
30 out.println("<p>The number of records in ArrayList are " +  
31 studentList.size() + "</p>");
```

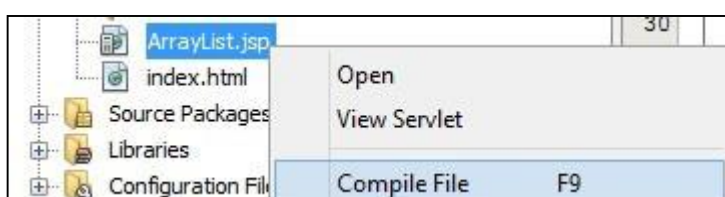
11. Finally, populate the list of students.

```
38  
39 //Populate a list of students..  
40 for (int i=0; i < studentList.size(); i++ )  
41 out.println("<p>Record " + (i+1) + " is " + studentList.get(i) + "</p>");
```

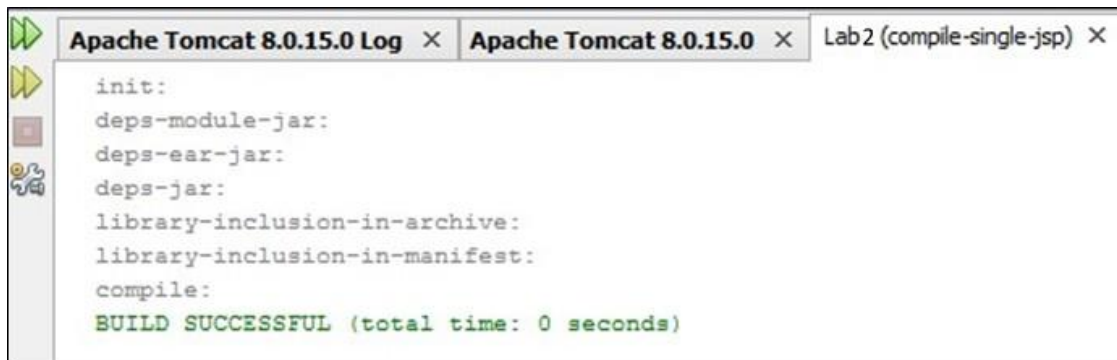
12. Click *SaveAll* icon.



13. Right click file *ArrayList.jsp* and click *Compile File (F9)*.

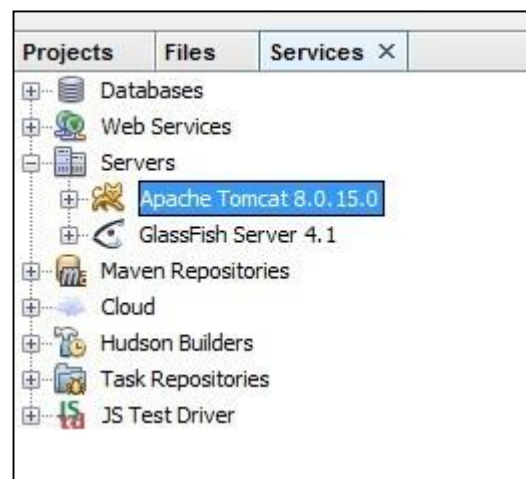


14. You will get notification message the the bottom of Netbeans IDE with the green colour.

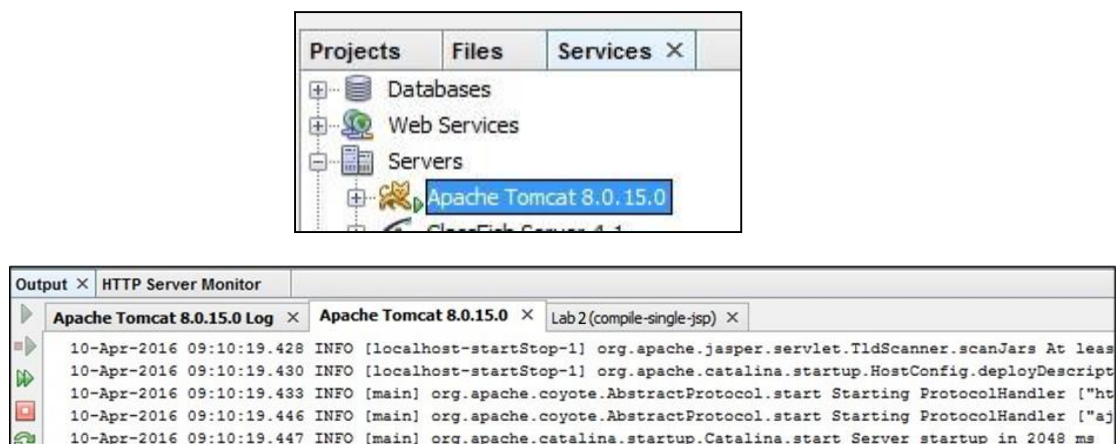


15. Before running any JSP's files for first time upon opening your Netbeans IDE, you need to start your web server (i.e; Apache Tomcat).

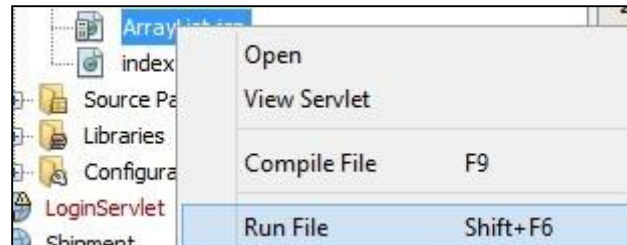
16. To perform this, go to *Services -> Servers -> Apache Tomcat*.



17. You should get the green indicator at *Apache Tomcat's* icon and *Apache Tomcat* output message with the time taken to start specified time to start *Apache Tomcat* web server.



18. Go to *Project's* tab. Then right click file *ArrayList.jsp* and click *Run File* (Shift+F6).



19. Output will appear in web browser.





## MY WORK

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ page import="java.util.ArrayList" %>
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Use Java ArrayList</title>
</head>
<body>
  <h1>Use JSP Page Directive</h1>
  <%
    // Create ArrayList object
    ArrayList<String> studentList = new ArrayList<String>();

    // Store student names
    studentList.add("Mohamad Azam");
    studentList.add("Peter Chong");
    studentList.add("Rahimah Mansor");
    studentList.add("Sri Devi");
    studentList.add("Ng Hue Ween");
    studentList.add("S. Nagarajan");

    // Display the number of records
    out.println("<p style='color: blue;'> The number of records in ArrayList are " +
      studentList.size() + "</p>");

    // Populate a list of students with blue font color
    for (int i = 0; i < studentList.size(); i++) {
      out.println("<p style='color: blue;'>Record " + (i + 1) + " is " + studentList.get(i) + "</p>");
    }
  %>
  <br/>
  <footer>&copy;2016-Mohamad Nor</footer>
</body>
</html>
```

## Use JSP Page Directive

The number of records in ArrayList are 6

Record 1 is Mohamad Azam

Record 2 is Peter Chong

Record 3 is Rahimah Mansor

Record 4 is Sri Devi

Record 5 is Ng Hue Ween

Record 6 is S. Nagarajan

©2016-Mohamad Nor



## Reflection

1. What you have learnt from this exercise?  
= how to make an array list using jsp
  
2. Write a sample syntax how you want to use java *Math* object in JSP?  
=  

```
double number = 25;  
double squareRoot = Math.sqrt(number);
```
  
3. List and write a sample syntax for THREE (3) of JSP page directive.
  - `<%@ include file="header.jsp" %>`
  
  - `<%@ page contentType="text/html; charset=UTF-8" %>`
  
  - `<%@ page contentType="text/html; charset=UTF-8" %>`

## Task 6: Use JSP Include directive for JSP Page

<b>Objective</b>	: Demonstrate the use of JSP Page Include directive.
<b>Problem</b>	: Create a JSP master page that displays the header, main contents and footer.
<b>Description</b>	
<b>Estimated time</b>	: 30 minutes

1. Create a new JSP's file.
2. Type file name as *mainPage*.
3. Create content for *mainPage.jsp* as below.

### Using JSP Include directive

Java Server Page (JSP) is a technology for controlling the content or appearance of Web pages through the use of servlets, small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it.

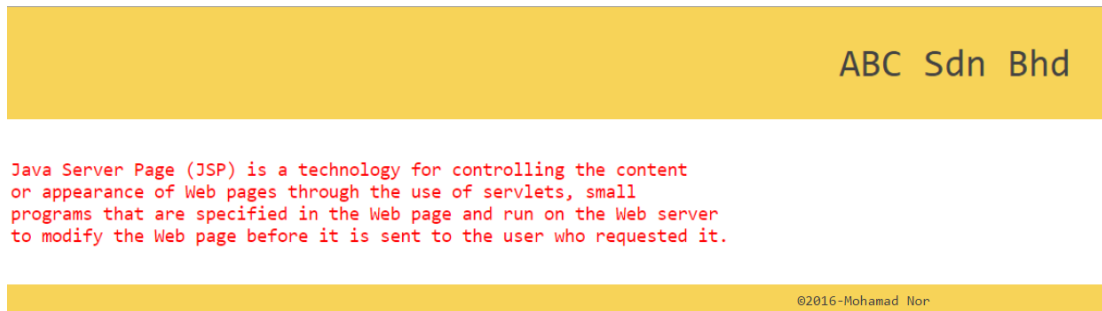
4. Create a header file as *headerPage.jsp* and display the following output.

ABC Sdn Bhd

5. Create a header file as *footerPage.jsp* and display the following output

©2016-Mohamad Nor

6. Include your *headerPage.jsp* and *footerPage.jsp* inside your *mainPage.jsp*.
7. Save *mainPage.jsp*
8. Compile and run *mainPage.jsp*.
9. You should get the following output.



## MY WORK

### mainPage.jsp

```
7 <%@ page contentType="text/html; charset=UTF-8" %>
8 <!DOCTYPE html>
9 <html>
10 <head>
11     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12     <title>JSP Page</title>
13 </head>
14 <body>
15     <%@ include file="headerPage.jsp" %>
16     <h1>Using JSP Include directive</h1>
17     <p style="color: red;">Java Server Page (JSP) is a technology for controlling the content
18         or appearance of web pages through the use of servlets, small
19         programs that are specified in the web page and run on the web server
20         to modify the web page before it is sent to the user who requested it.</p>
21     <%@ include file="footerPage.jsp" %>
22 </body>
23 </html>
24
```

## Using JSP Include directive

Java Server Page (JSP) is a technology for controlling the content or appearance of web pages through the use of servlets, small programs that are specified in the web page and run on the web server to modify the web page before it is sent to the user who requested it.

©2016-Mohamad Nor

### headerPage.jsp

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Header Page</title>
  <style>
    .header-container {
      background-color: yellow;
      text-align: center;
      padding: 10px;
    }
    .header-text {
      font-size: 50px;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div class="header-container">
    <span class="header-text">ABC Sdn Bhd</span>
  </div>
</body>
</html>
```

## footerPage.jsp

```
<%@ page contentType="text/html; charset=UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Header Page</title>
  <style>
    .header-container {
      background-color: yellow;
      text-align: center;
      padding: 10px;
    }
    .header-text {
      font-size: 24px;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div class="header-container">
    <span class="header-text">&copy;2016-Mohamad Nor</span>
  </div>
</body>
</html>
```

### Reflection

1. What you have learnt from this exercise?

= how to combine many jsp in one main jsp page.

2. Write a syntax how you want to include *common.html* file that located at a directory known as *master*.

= <%@ include file="/master/common.html" %>

## Exercise

1. Write a JSP's page to convert temperatures to Fahrenheit temperatures and via versa. The formula is given as:

$$F = (9/5)C + 32$$

Your program should ask the user to enter a temperature in Celsius, and then display the temperature converted to Fahrenheit.

## MY WORK

```
7 <%@ page contentType="text/html; charset=UTF-8" %>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>Temperature Converter</title>
13 </head>
14 <body>
15 <h1>Temperature Converter</h1>
16 <form method="post" action="calculateTemperature.jsp">
17 <label for="temperature">Enter temperature:</label>
18 <input type="number" id="temperature" name="temperature" required>
19 <select name="unit" required>
20 <option value="Celsius">Celsius</option>
21 <option value="Fahrenheit">Fahrenheit</option>
22 </select>
23 <input type="submit" value="Convert">
24 </form>
25 <%
26 // Check if the form was submitted
27 if (request.getMethod().equalsIgnoreCase("post")) {
28 double temperature = Double.parseDouble(request.getParameter("temperature"));
29 String unit = request.getParameter("unit");
30 double convertedTemperature;
31
32 if (unit.equalsIgnoreCase("Celsius")) {
33 convertedTemperature = (9.0 / 5.0) * temperature + 32;
34 out.println("<p>" + temperature + " Celsius is " + convertedTemperature + " Fahrenheit</p>");
35 } else if (unit.equalsIgnoreCase("Fahrenheit")) {
36 convertedTemperature = (temperature - 32) * 5.0 / 9.0;
37 out.println("<p>" + temperature + " Fahrenheit is " + convertedTemperature + " Celsius</p>");
38 }
```

Temperature Converter

Enter temperature:  Celsius

25.0 Fahrenheit is -3.888888888888889 Celsius

- Write a JSP's form that asks for the length and width of two rectangles. The program should tell the user which rectangle has the greater area, or if the areas are the same. [Note: All result must be in 2 decimal places].

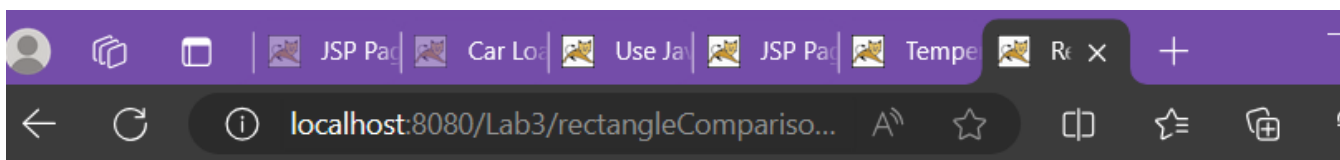
## MY WORK

```

7  <%@ page contentType="text/html; charset=UTF-8" %>
8  <!DOCTYPE html>
9  <html>
10 <head>
11     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12     <title>Rectangle Area Comparison</title>
13 </head>
14 <body>
15     <h1>Rectangle Area Comparison</h1>
16     <form method="post" action="rectangleComparison.jsp">
17         <label for="rect1Length">Rectangle 1 Length:</label>
18         <input type="number" id="rect1Length" name="rect1Length" step="0.01" required>
19         <label for="rect1Width">Rectangle 1 Width:</label>
20         <input type="number" id="rect1Width" name="rect1Width" step="0.01" required>
21         <br>
22         <label for="rect2Length">Rectangle 2 Length:</label>
23         <input type="number" id="rect2Length" name="rect2Length" step="0.01" required>
24         <label for="rect2Width">Rectangle 2 Width:</label>
25         <input type="number" id="rect2Width" name="rect2Width" step="0.01" required>
26         <br>
27         <input type="submit" value="Compare">
28     </form>
29     <%
30         if (request.getMethod().equalsIgnoreCase("post")) {
31             double rect1Length = Double.parseDouble(request.getParameter("rect1Length"));
32             double rect1Width = Double.parseDouble(request.getParameter("rect1Width"));
33             double rect2Length = Double.parseDouble(request.getParameter("rect2Length"));
34             double rect2Width = Double.parseDouble(request.getParameter("rect2Width"));
35
36             double area1 = rect1Length * rect1Width;
37             double area2 = rect2Length * rect2Width;

```

```
Source History
25 <input type="number" id="rect2Width" name="rect2Width" step="0.01" required>
26 <br>
27 <input type="submit" value="Compare">
28 </form>
29 <%
30     if (request.getMethod().equalsIgnoreCase("post")) {
31         double rect1Length = Double.parseDouble(request.getParameter("rect1Length"));
32         double rect1Width = Double.parseDouble(request.getParameter("rect1Width"));
33         double rect2Length = Double.parseDouble(request.getParameter("rect2Length"));
34         double rect2Width = Double.parseDouble(request.getParameter("rect2Width"));
35
36         double area1 = rect1Length * rect1Width;
37         double area2 = rect2Length * rect2Width;
38
39         out.println("<p>Area of Rectangle 1: " + String.format("%.2f", area1) + " square units</p>");
40         out.println("<p>Area of Rectangle 2: " + String.format("%.2f", area2) + " square units</p>");
41
42         if (area1 > area2) {
43             out.println("<p>Rectangle 1 has a greater area than Rectangle 2.</p>");
44         } else if (area1 < area2) {
45             out.println("<p>Rectangle 2 has a greater area than Rectangle 1.</p>");
46         } else {
47             out.println("<p>Both rectangles have the same area.</p>");
48         }
49     }
50 <%>
51 </body>
52 </html>
53
54
```



## Rectangle Area Comparison

Rectangle 1 Length:	<input type="text"/>	Rectangle 1 Width:	<input type="text"/>
Rectangle 2 Length:	<input type="text"/>	Rectangle 2 Width:	<input type="text"/>
<input type="button" value="Compare"/>			

Area of Rectangle 1: 20.00 square units

Area of Rectangle 2: 145.00 square units

Rectangle 2 has a greater area than Rectangle 1.