UNIVERSITI MALAYSIA TERENGGANU

CSM3023

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

LAB 3

SEMESTER II 2023/2024

Prepared for:

DR. MOHAMMAD NOR HASSAN

Prepared by:

HUSNA ZAHIRA BINTI RUZELI (S67554)
(K4)

Week 2

JSP: Scriplet, Page Directive & Include Directive

Web Programming 2

Name:
HUSNA
ZAHIRA
BT RUZELI

Matric #:
S67554

Semester:
II
2023/202
4

Lab: MP1

Demonstrator: ENCIK 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

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’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.

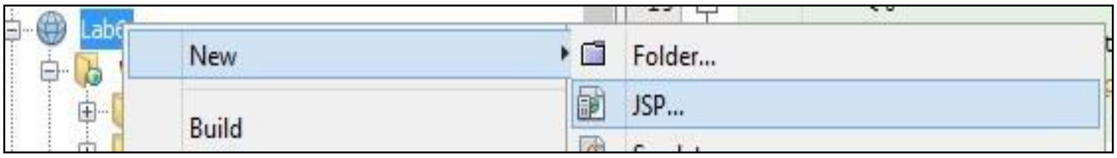
4. Create a page *memberProcessing.jsp*.

5. When user click *Submit* button, process the request and display the input key-in in *memberProcessing.jsp* page.

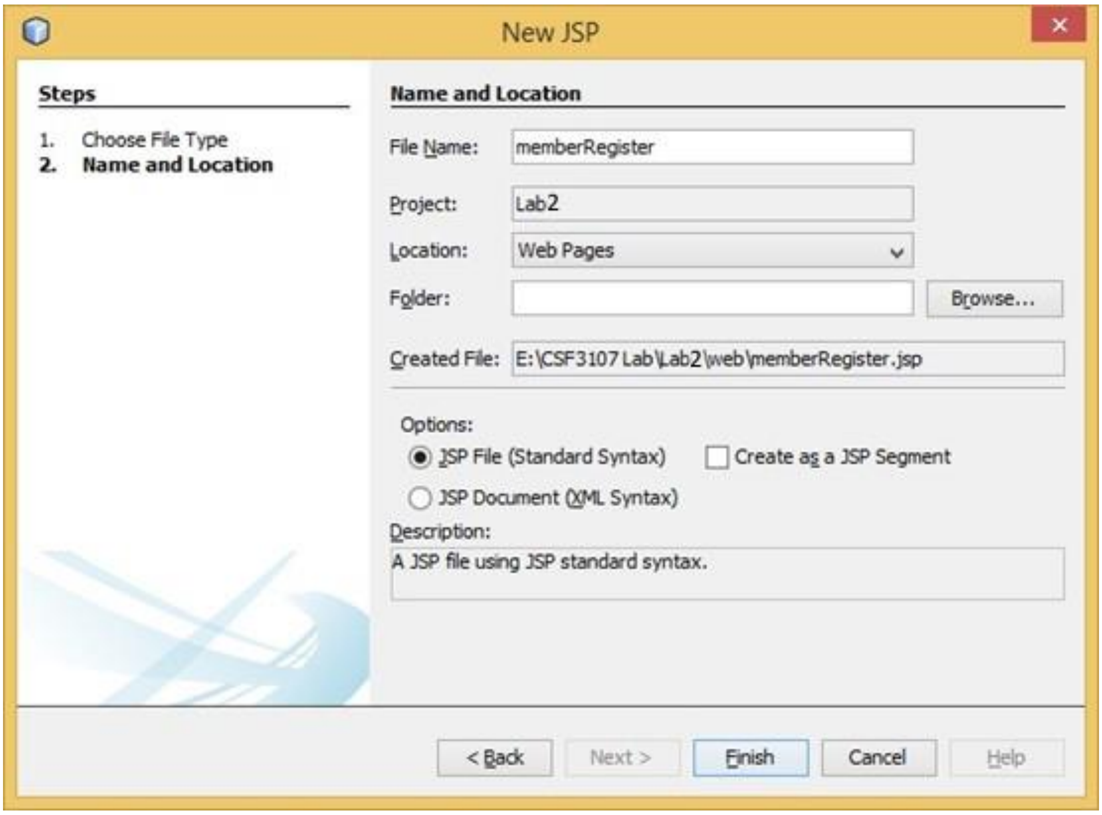
Estimated time : 30 minutes

1. Create new Project namely *Lab2*.

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



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



File Name: memberRegister

Project: Lab2

Location: Web Pages

Folder:

Created File: E:\CSF3107 Lab\Lab2\web\memberRegister.jsp

Options:

● JSP File (Standard Syntax) □ Create as a JSP Segment

○ JSP Document (XML Syntax)

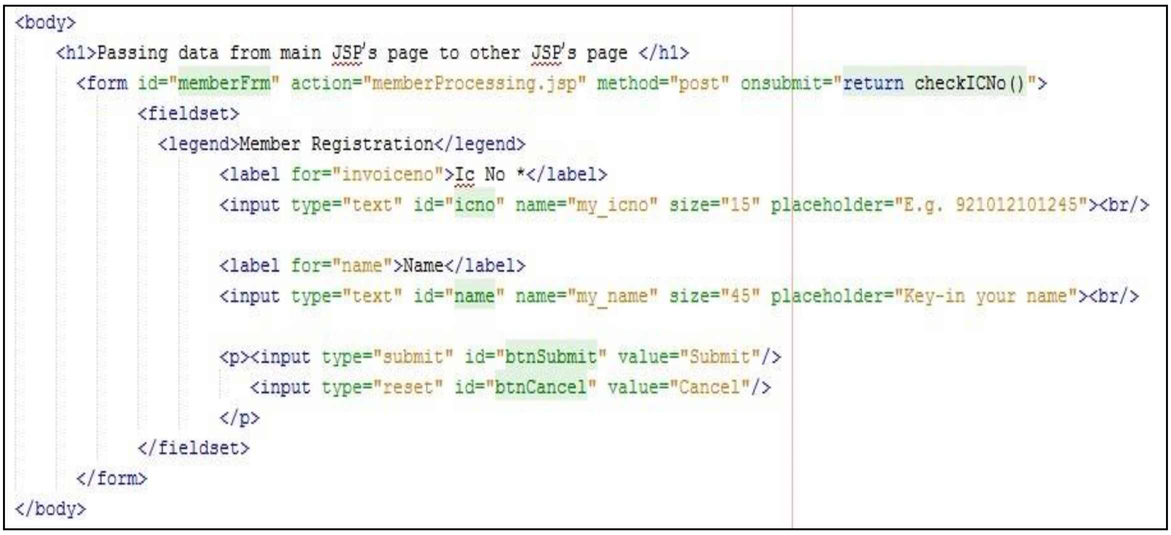
Description:

A JSP file using JSP standard syntax.

4. Click *Finish* button.

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

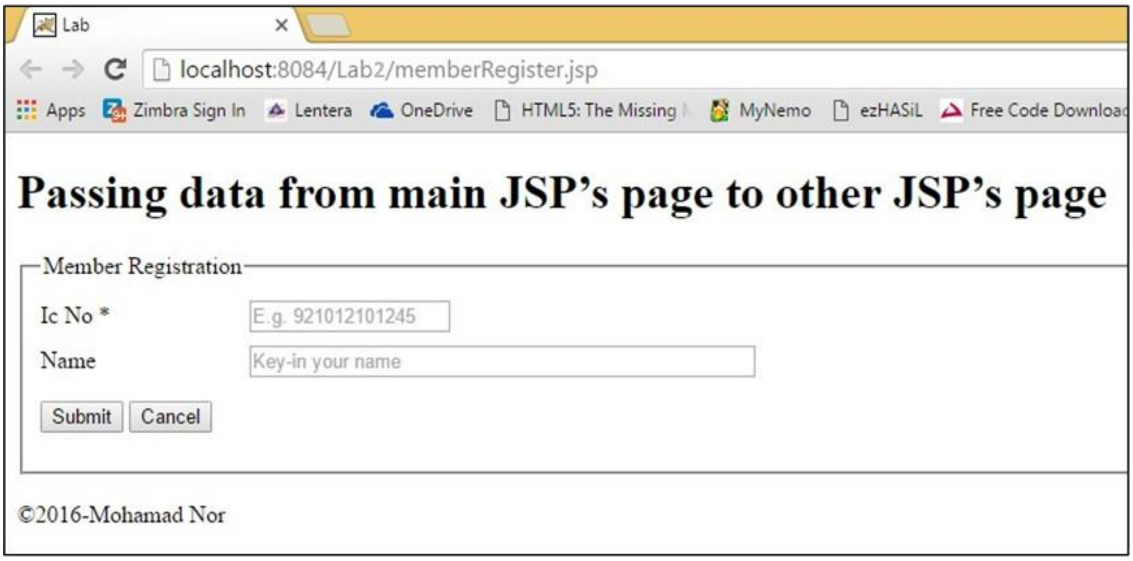
6. Write a HTML's markup to produce HTML's form



```
<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:



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

Member Registration

Ic No * E.g. 921012101245

Name Key-in your name

Submit Cancel

©2016-Mohamad Nor

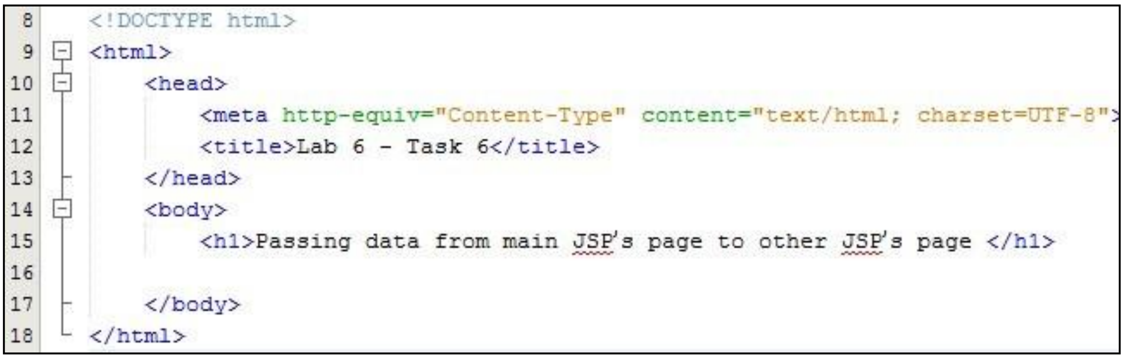
9. Repeat step 1 and step 2.

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

11. Click *Finish* button.

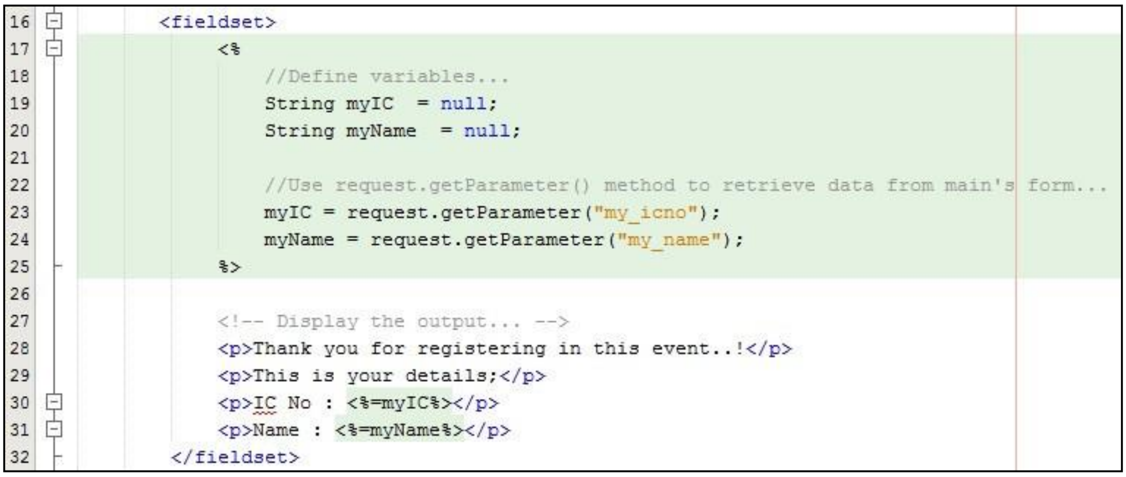
12. Source code for *memberProcessing.jsp* will appear.

13. Write a HTML code.



```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Lab 6 - Task 6</title>
    </head>
    <body>
        <h1>Passing data from main JSP's page to other JSP's page </h1>

    </body>
</html>
```

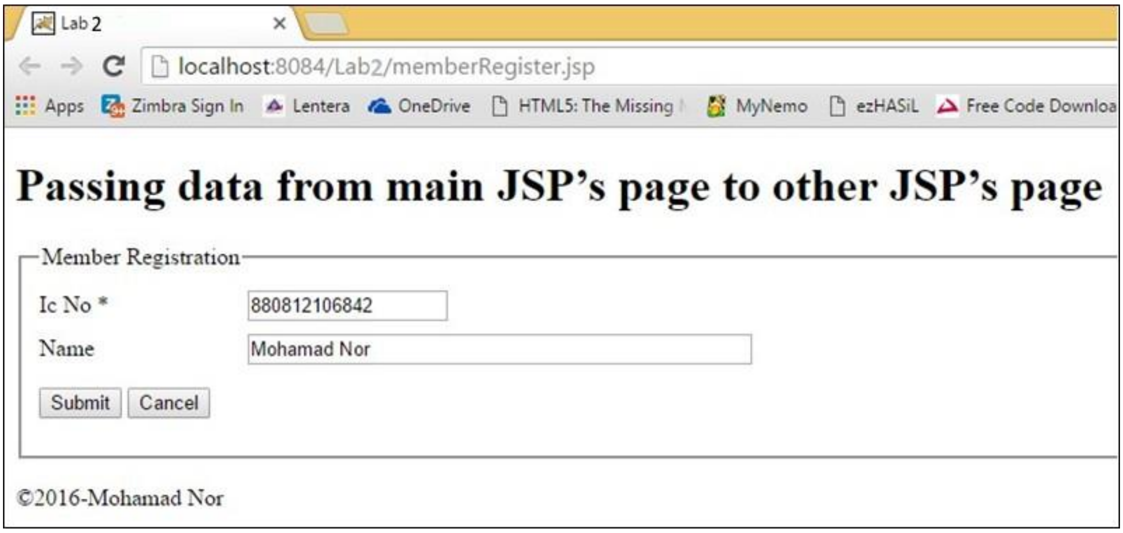
14. Add additional HTML’s tag and Java Scriplet to retrieve the value from main’s form.



```
  <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>
```

15. Compile *memberProcessing.jsp* file.

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



Passing data from main JSP’s page to other JSP’s page

Member Registration

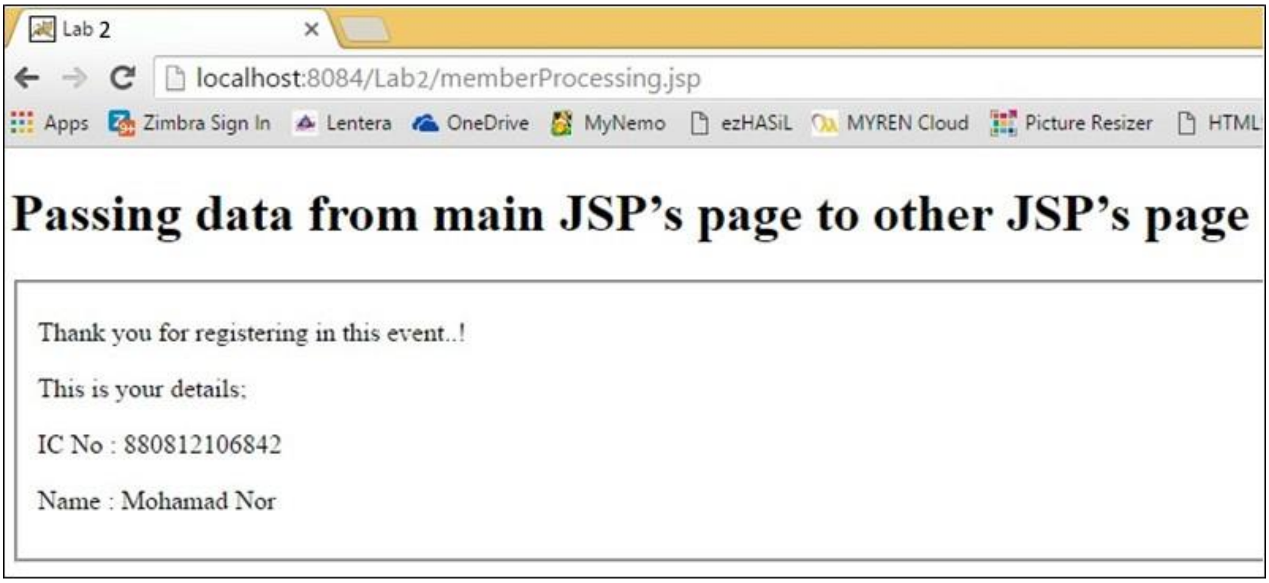
Ic No * 880812106842

Name Mohamad Nor

©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.



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?

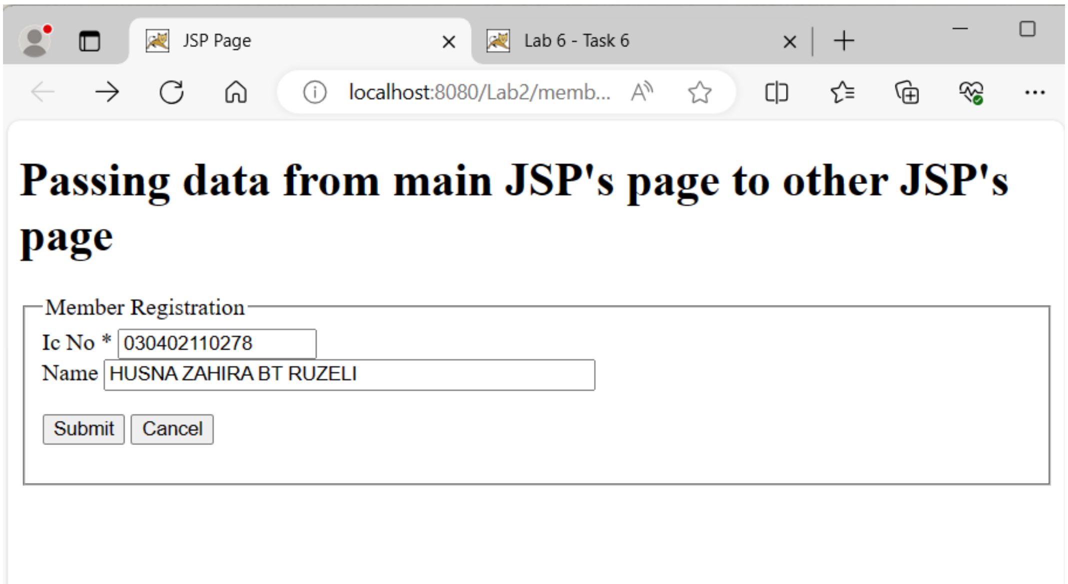
   Answer:

   Using session variables method, hidden fields method, or Javascript.

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?

Answer:
It will display null.

Output:



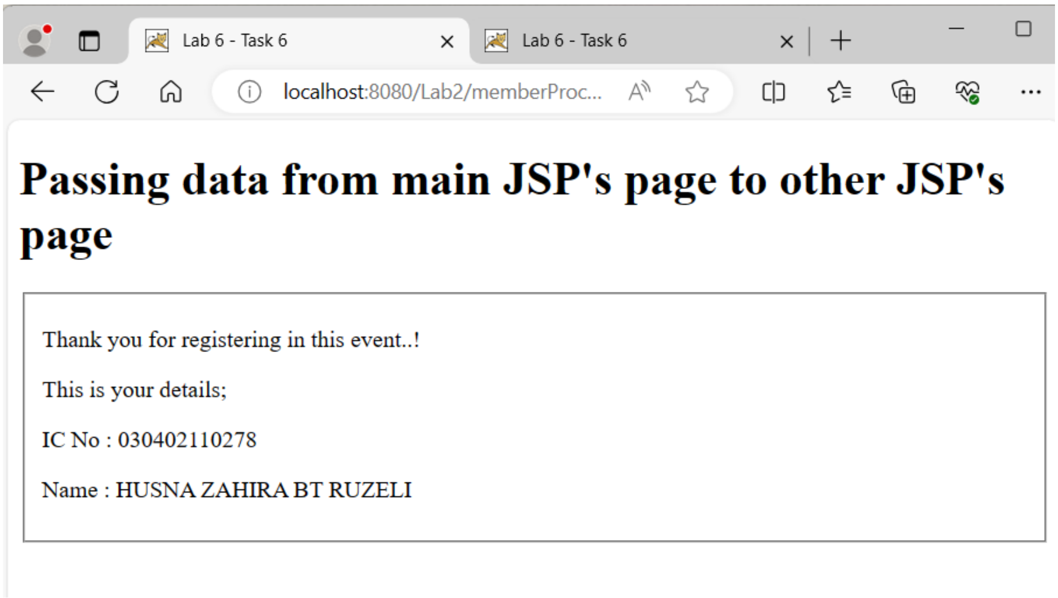
Passing data from main JSP's page to other JSP's page

Member Registration

Ic No * 030402110278

Name HUSNA ZAHIRA BT RUZELI

Submit Cancel



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 : 030402110278

Name : HUSNA ZAHIRA BT RUZELI

Task 2: Using Mathematics Operations in JSP

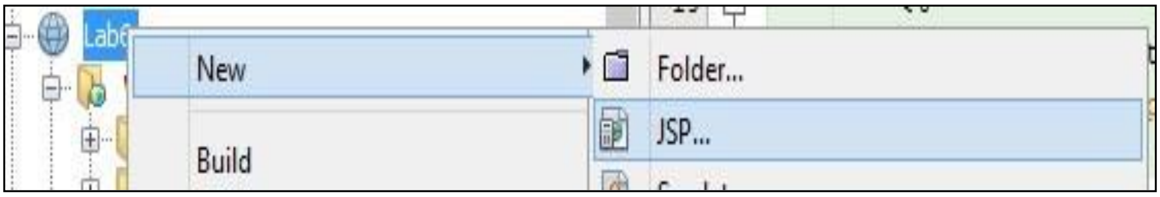
Objective : To demonstrate the use of *request.getParameter (“Mathematics operations”)* in JSP’s page.

Problem Description : i. Create a page *Calculator.jsp* consists of interface represent basic calculator.

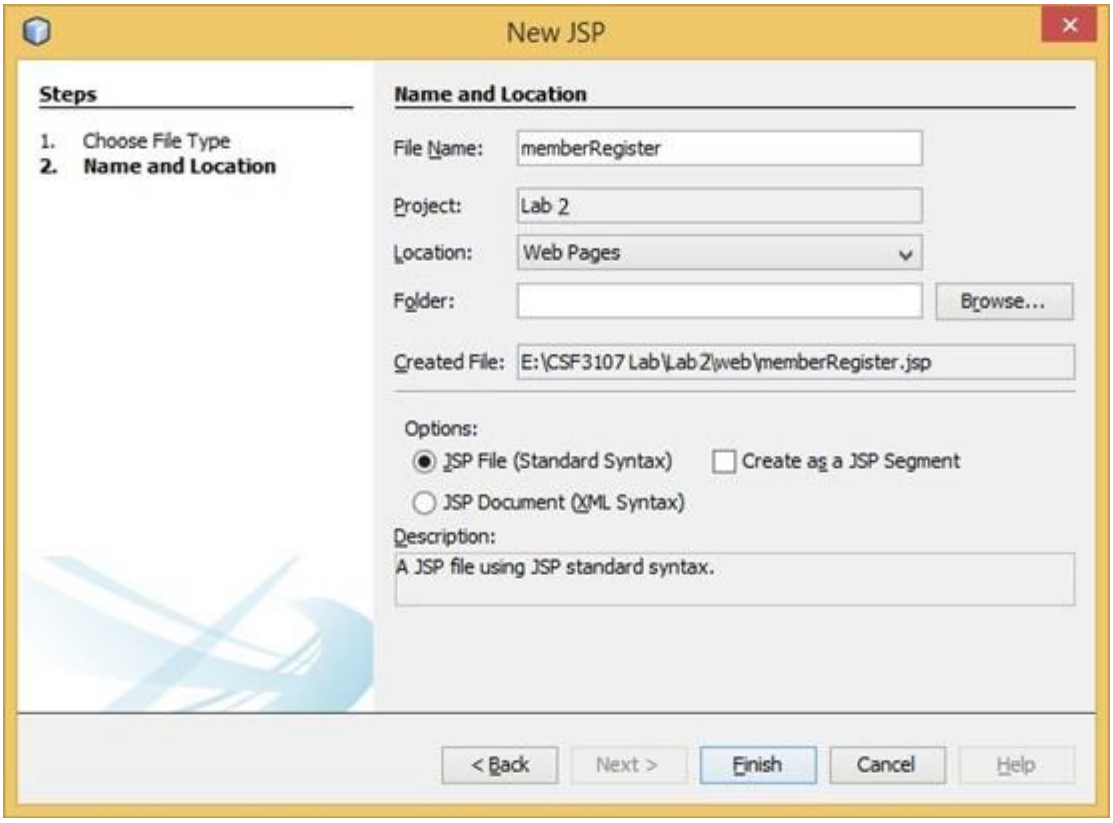
ii. When user key-in inputs, process the request and display the results direct in JSP page.

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*.



New JSP

Steps

1. Choose File Type
2. Name and Location

Name and Location

File Name: memberRegister

Project: Lab 2

Location: Web Pages

Folder:

Created File: E:\CSF3107 Lab\Lab 2\web\memberRegister.jsp

Options:

● JSP File (Standard Syntax) □ Create as a JSP Segment

○ JSP Document (XML Syntax)

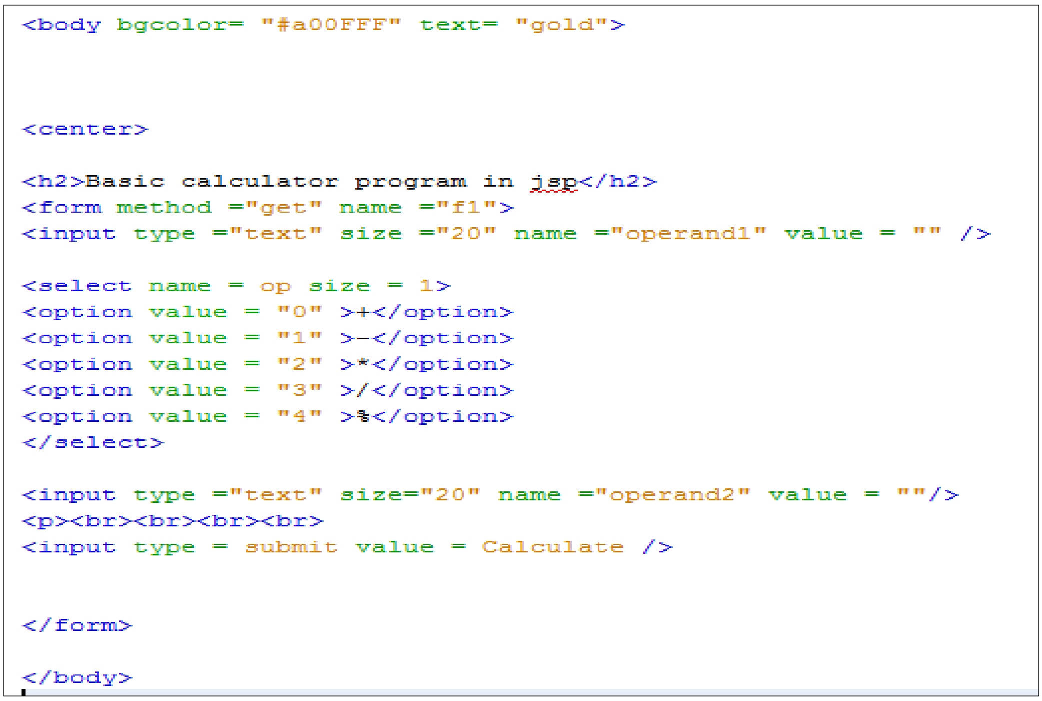
Description:

A JSP file using JSP standard syntax.

4. Click *Finish* button.

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

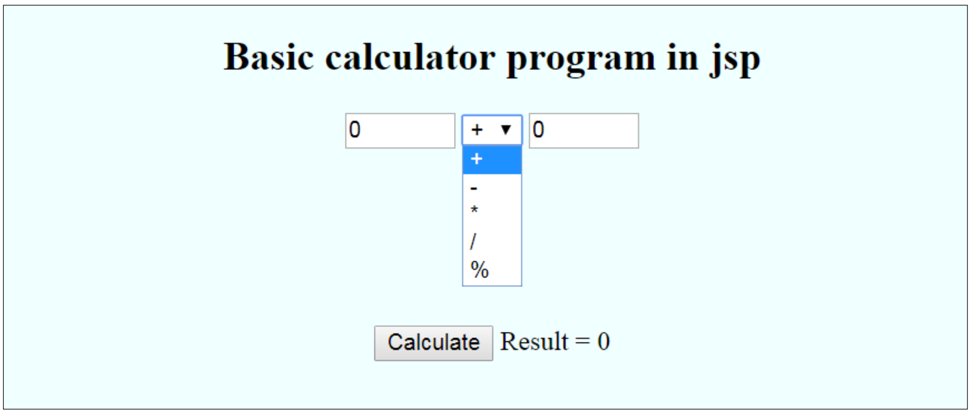
6. Write a HTML's markup to produce HTML's form



```
<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.

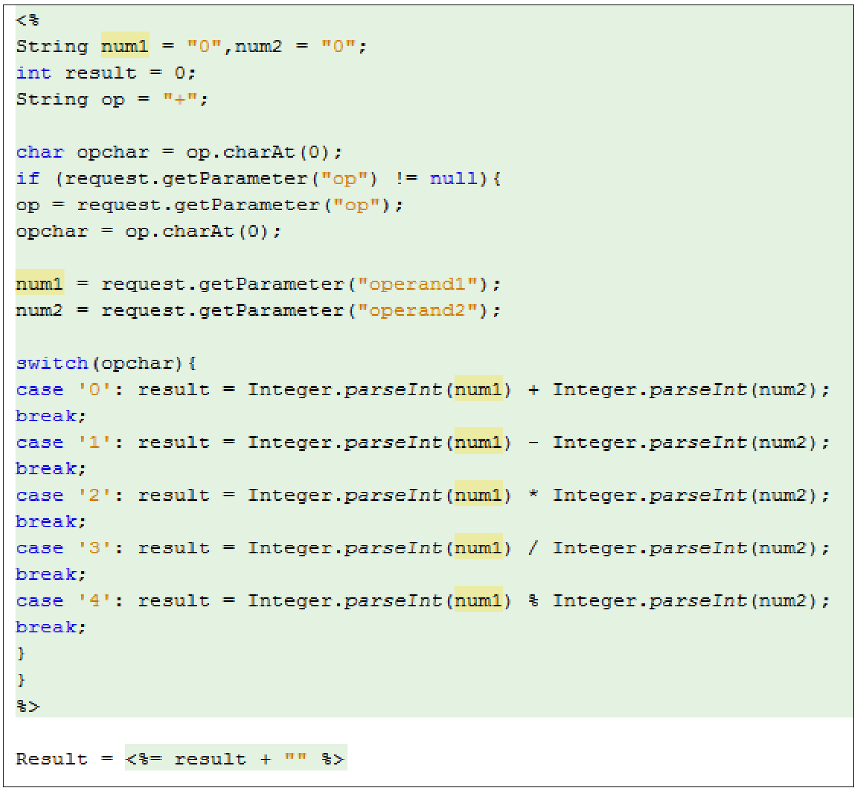
8. Run the *Calculator.jsp* file and you should get the interface as below:



Basic calculator program in jsp

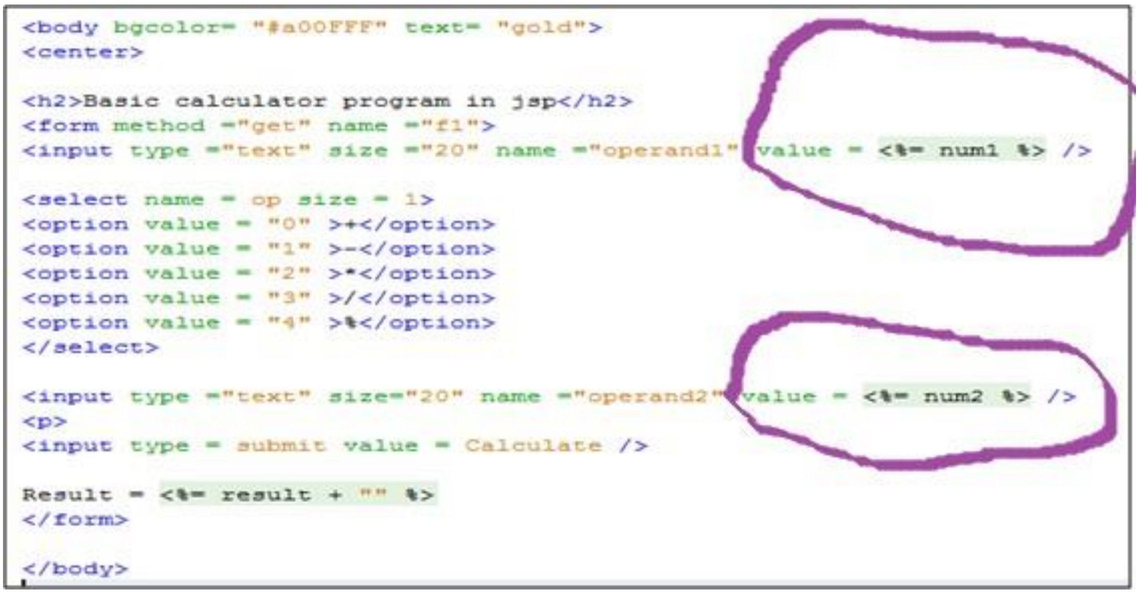
Result = 0

9. Add additional HTML's tag and Java scriplet 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.

Reflection

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

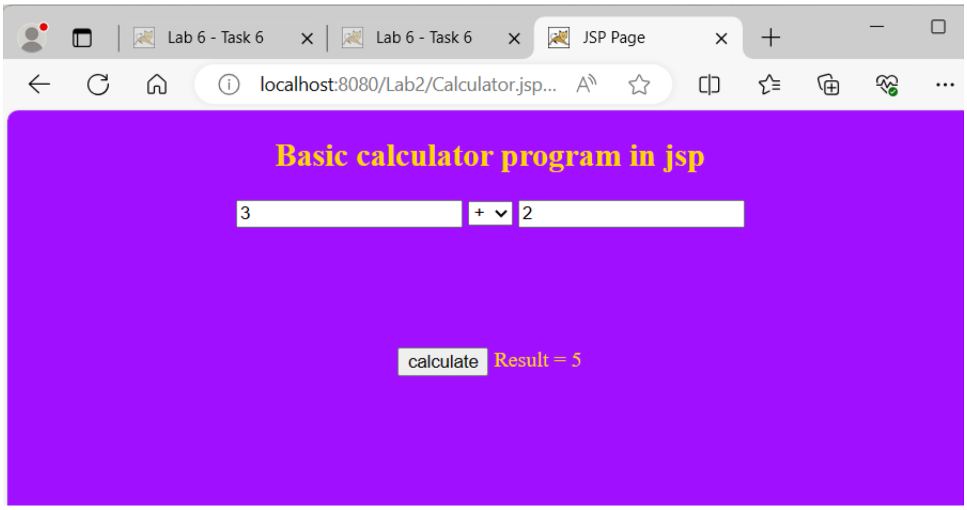
Answer:

Using session variables method, hidden fields method, or Javascript.

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?

Answer:
It will display null.

Output:



Basic calculator program in jsp

Result = 5

Task 3: Populate an Array Values into HTML’s Table

Objective : Read Java array and populate it into HTML’s table.

Problem Description : i. Create 2D array that store sales data.

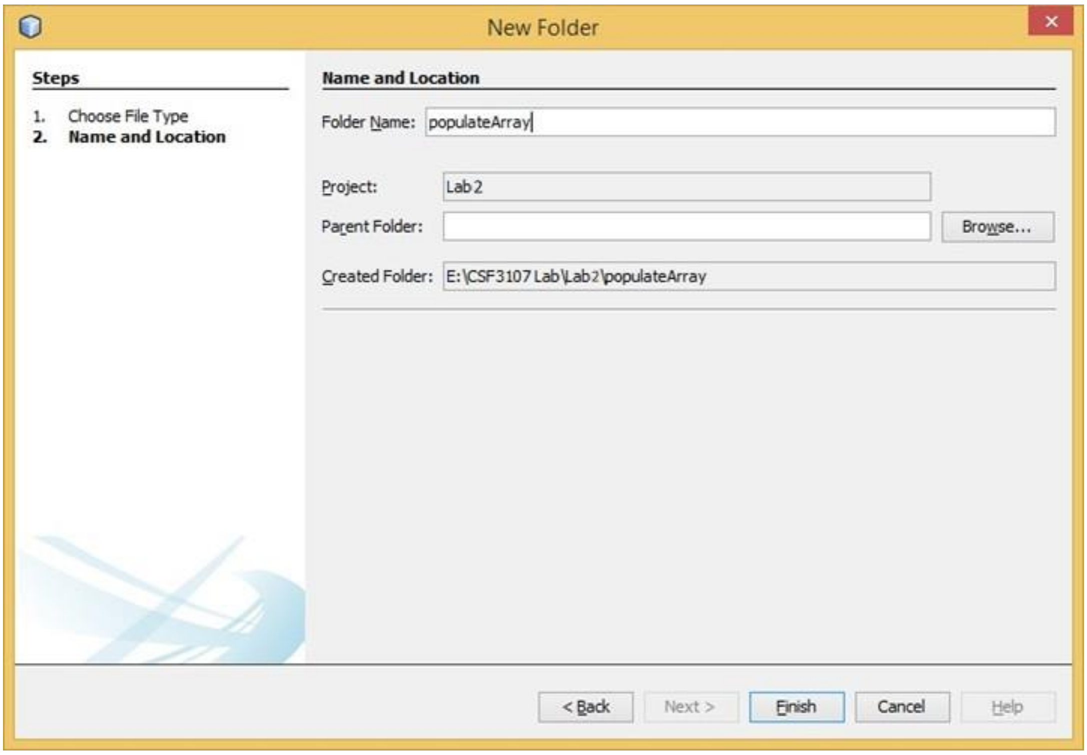
ii. Then, read an array and populate into HTML’s table.

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*.



New Folder

Steps

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

Name and Location

Folder Name: populateArray

Project: Lab2

Parent Folder:

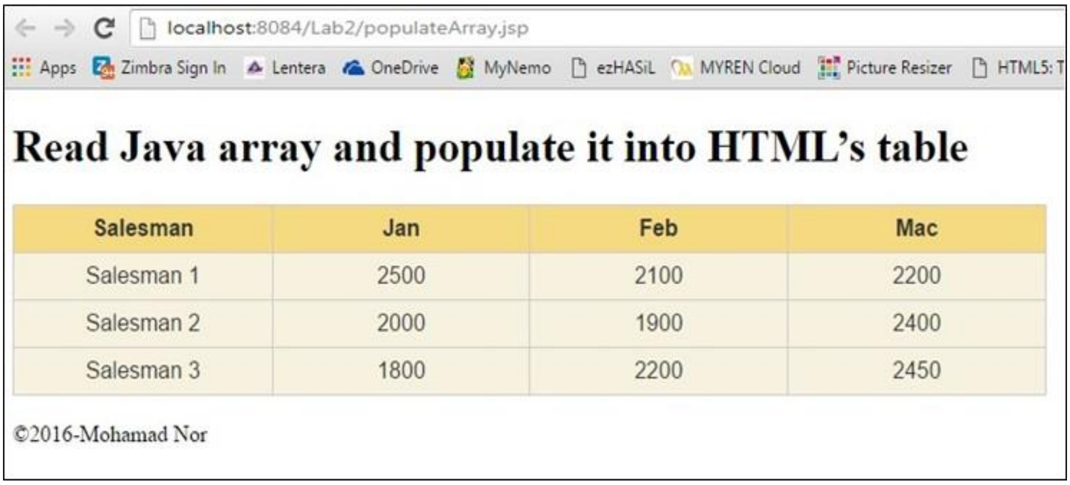
Created Folder: E:\CSF3107 Lab\Lab2\populateArray

4. Click *Finish* button.

5. Prepare standard HTML's a markup for page *populateArray.jsp*.
6. Write a Java Scriplet 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:



Read Java array and populate it into HTML's table

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

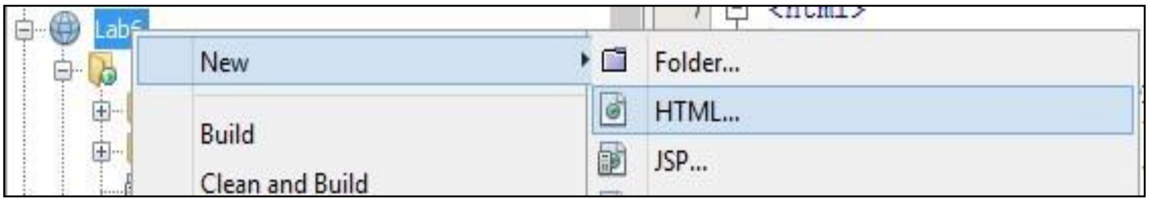
Objective : Passing input to next page for further processing.

Problem Description : 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.

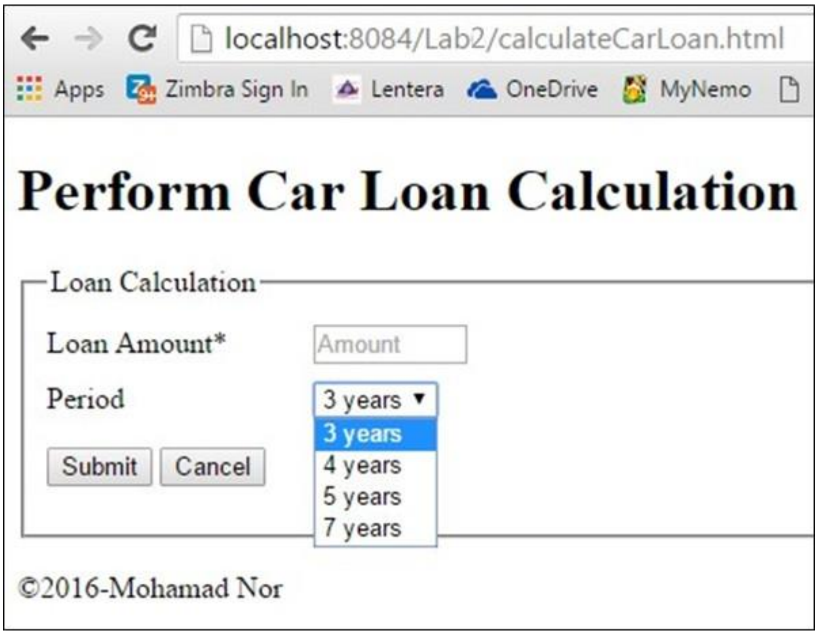
Estimated time : 50 minutes

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.



Perform Car Loan Calculation

Loan Calculation

Loan Amount* Amount

Period 3 years

©2016-Mohamad Nor

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

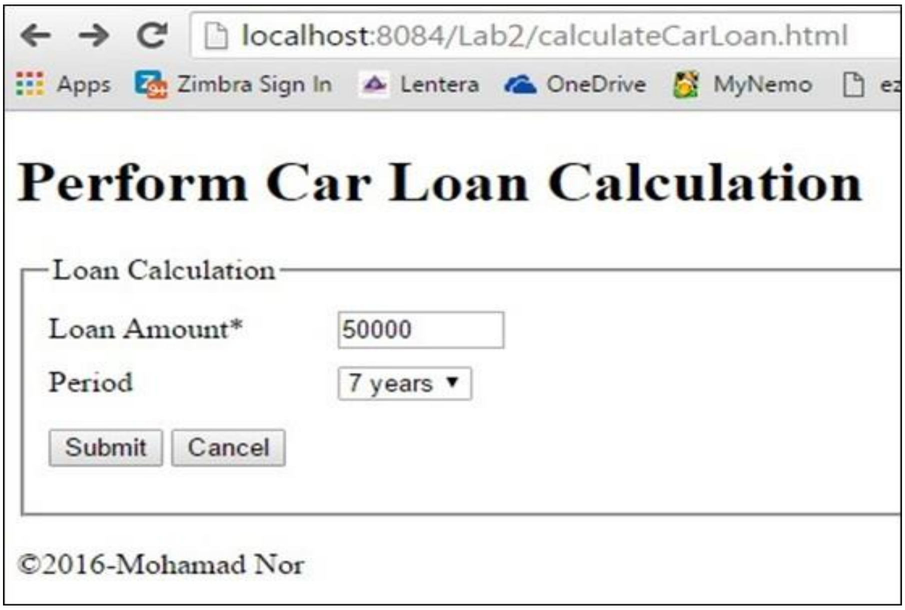
9. Construct the logic for calculating car loan and display the result.

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;



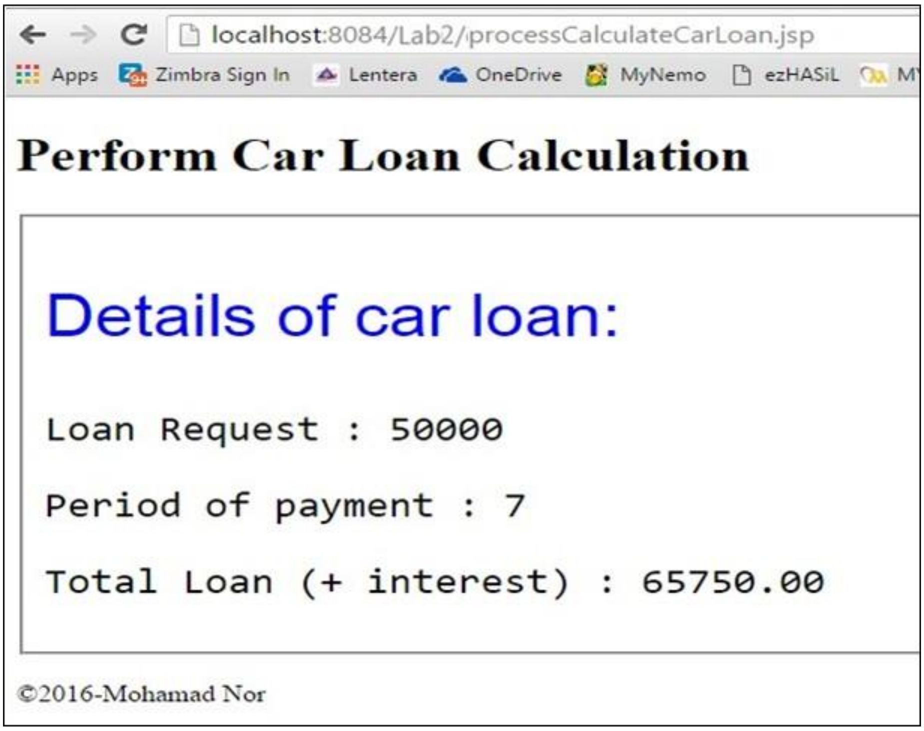
Perform Car Loan Calculation

Loan Calculation

Loan Amount* 50000

Period 7 years

©2016-Mohamad Nor



localhost:8084/Lab2/processCalculateCarLoan.jsp

Perform Car Loan Calculation

Details of car loan:

Loan Request : 50000

Period of payment : 7

Total Loan (+ interest) : 65750.00

©2016-Mohamad Nor

Reflection

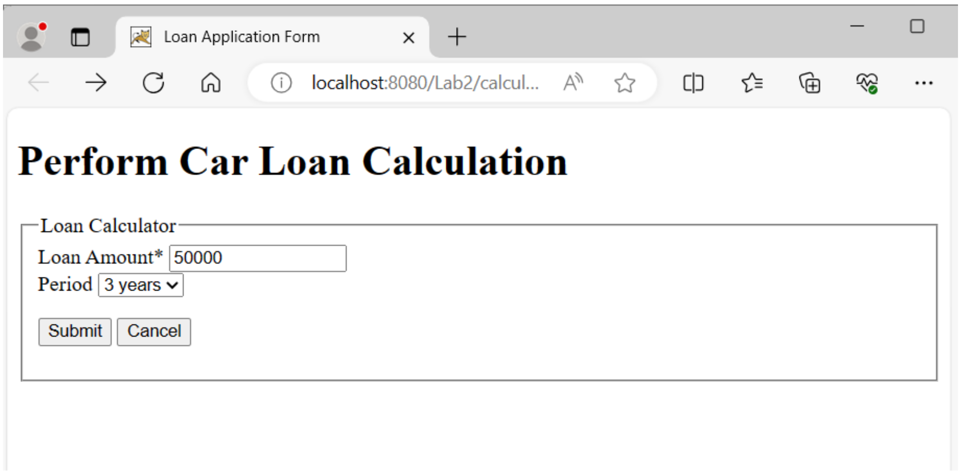
1. How you want to retrieve data from previous page?

   Answer:

   Link the jsp file to <form action> tag in HTML file

2. Where the construction of logic occur for calculating Total Loan (+ interest) ?
   Answer:
   In the <% tag in processCalculateCarLoan.jsp file

Output:

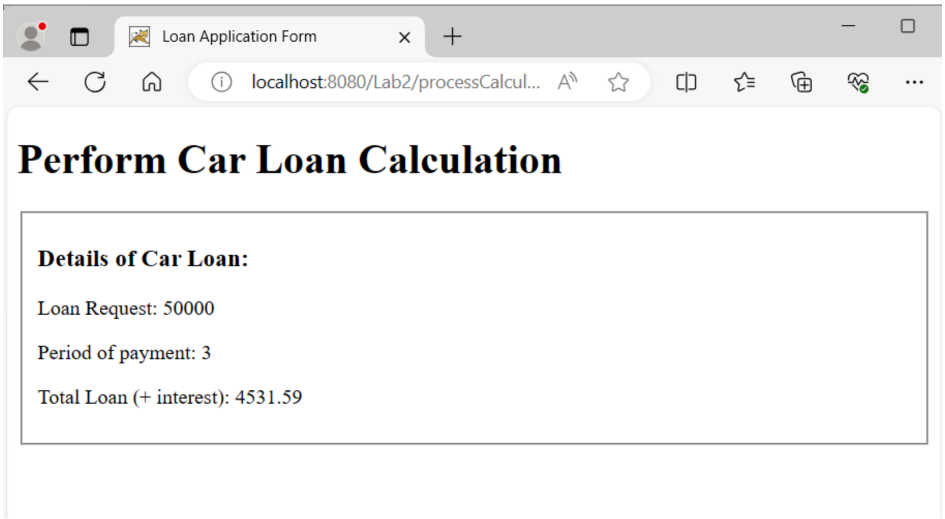


localhost:8080/Lab2/calcul...

Perform Car Loan Calculation

Loan Calculator

Loan Amount*	50000
Period	3 years



Perform Car Loan Calculation

Details of Car Loan:

Loan Request: 50000

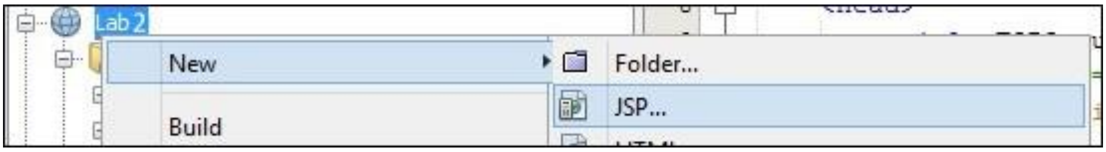
Period of payment: 3

Total Loan (+ interest): 4531.59

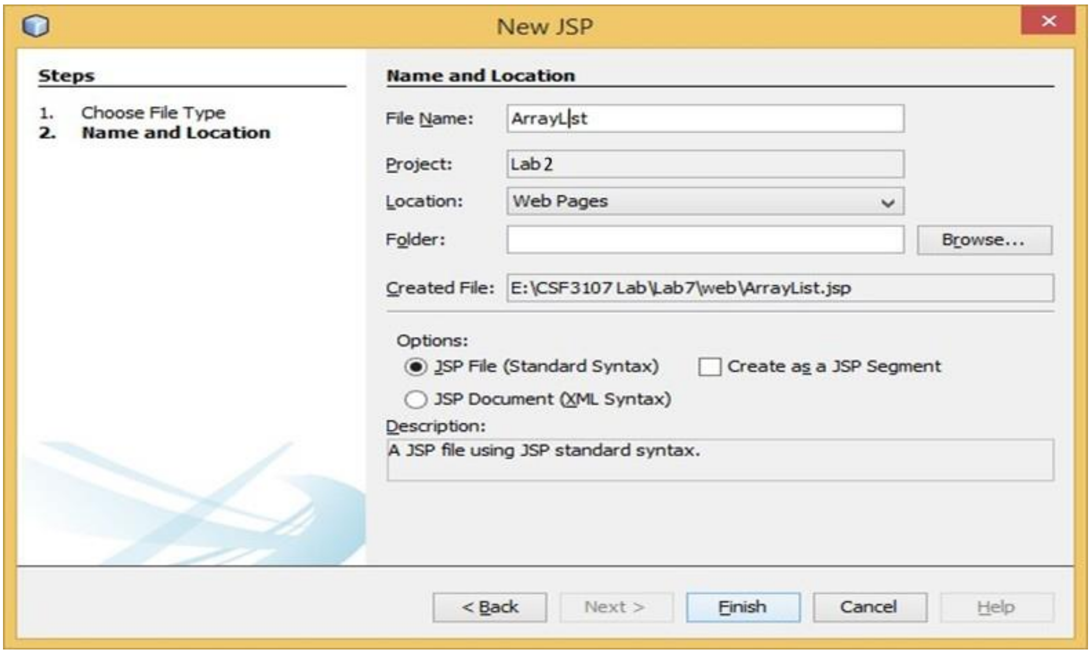
Task 5: Using JSP Page Directive to Call Java API

Objective	:	Use JSP page directive elements to call certain Java API.
Problem Description	:	Using Java *ArrayList* object to store data and retrive it via JSP page.
Estimated time	:	20 minutes

1. Create a new JSP's file.



2. Type file name as *ArrayList*.



File Name: ArrayList

Project: Lab 2

Location: Web Pages

Folder:

Created File: E:\CSF3107 Lab\Lab7\web\ArrayList.jsp

Options:

● JSP File (Standard Syntax) □ Create as a JSP Segment

○ JSP Document (XML Syntax)

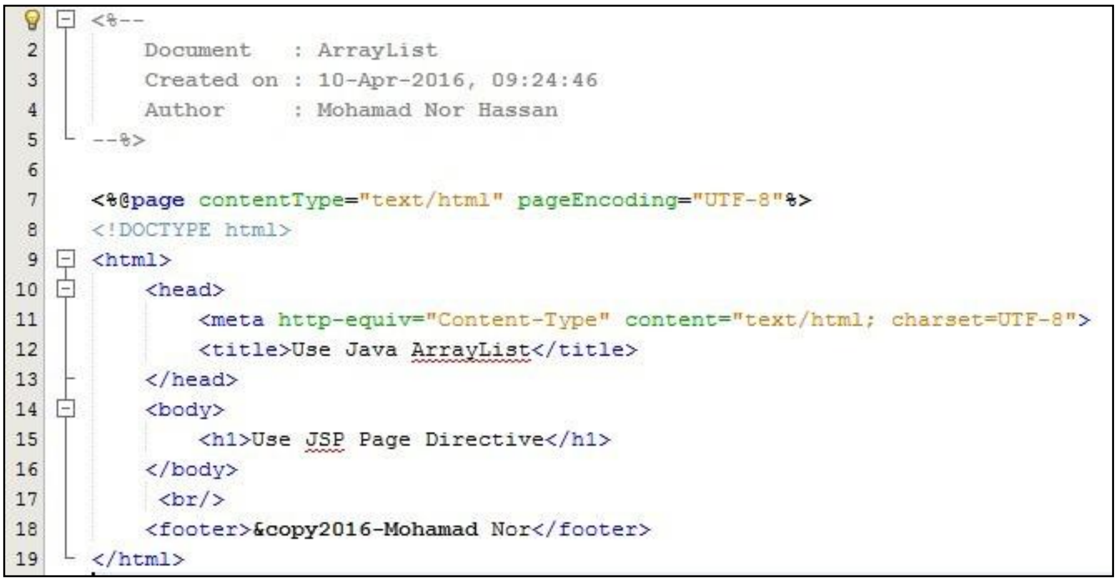
Description:

A JSP file using JSP standard syntax.

2. Click *Finish* button.

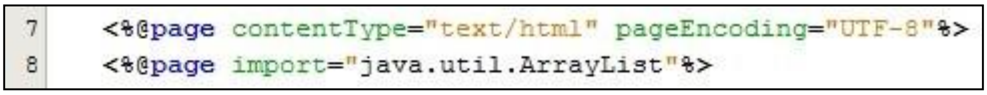
4. Type title as *Use Java ArrayList*.

5. Type header1 as *Use JSP Page Directive*



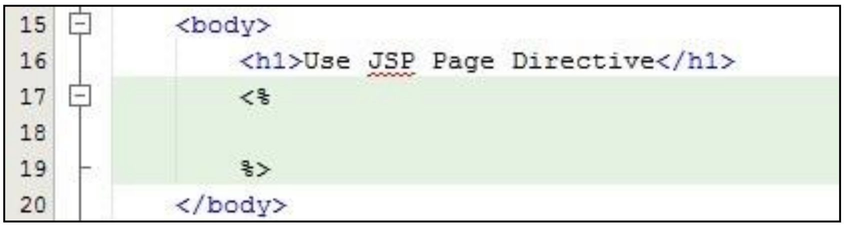
```
<%--
    Document   : ArrayList
    Created on : 10-Apr-2016, 09:24:46
    Author     : Mohamad Nor Hassan
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!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>
    </body>
     <br/>
    <footer>&copy2016-Mohamad Nor</footer>
</html>
```

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



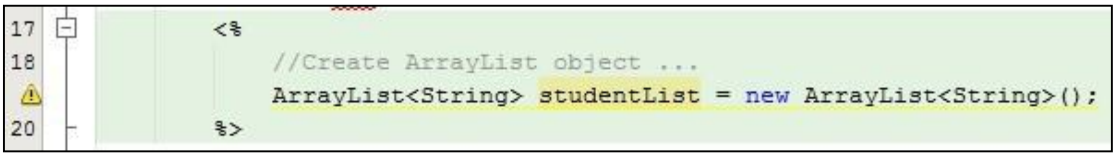
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.ArrayList"%>
```

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



```
    <body>
        <h1>Use JSP Page Directive</h1>
        <%

        %>
    </body>
```

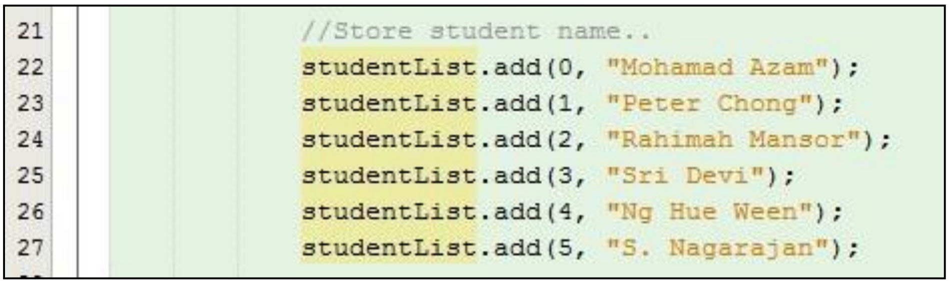
8. Create an object *ArrayList* to store a list of student name.



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

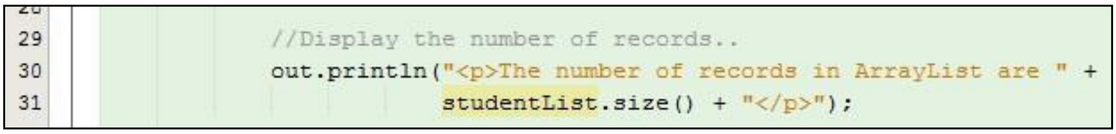
9. Add the following name to ArrayList's object.

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



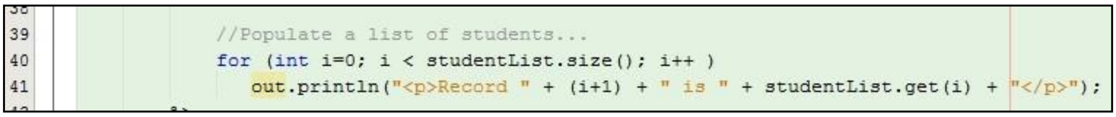
```
//Store student name..
studentList.add(0, "Mohamad Azam");
studentList.add(1, "Peter Chong");
studentList.add(2, "Rahimah Mansor");
studentList.add(3, "Sri Devi");
studentList.add(4, "Ng Hue Ween");
studentList.add(5, "S. Nagarajan");
```

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



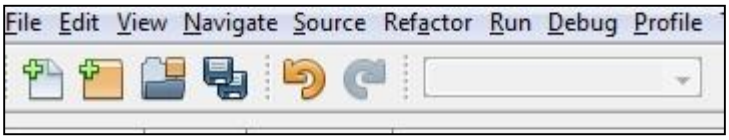
```
//Display the number of records..
out.println("<p>The number of records in ArrayList are " +
        studentList.size() + "</p>");
```

11. Finally, populate the list of students.

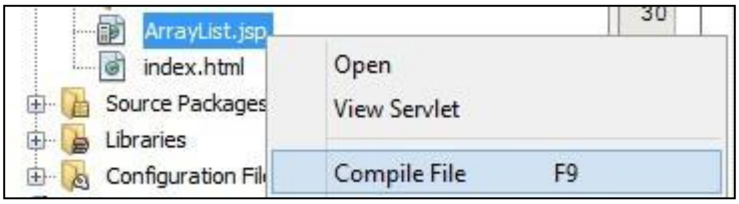


```
//Populate a list of students...
for (int i=0; i < studentList.size(); i++ )
   out.println("<p>Record " + (i+1) + " is " + studentList.get(i) + "</p>");
```

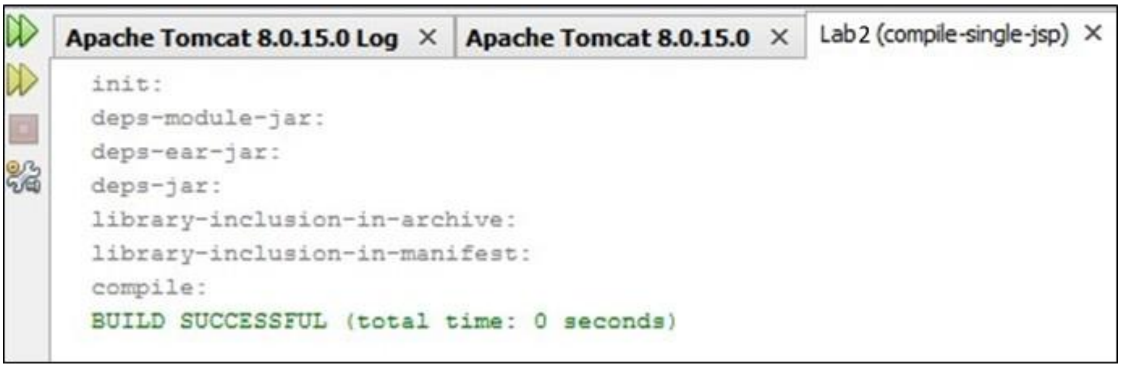
12. Click *SaveAll* icon.



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



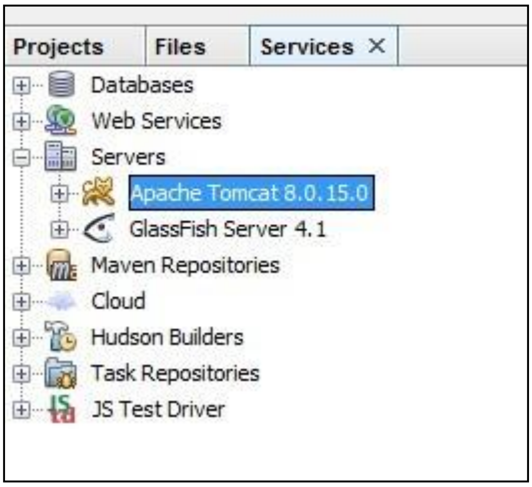
14. You will get notification message the the bottom of Netbeans IDE with the green colour.



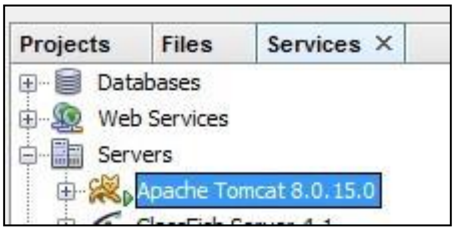
```
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
BUILD SUCCESSFUL (total time: 0 seconds)
```

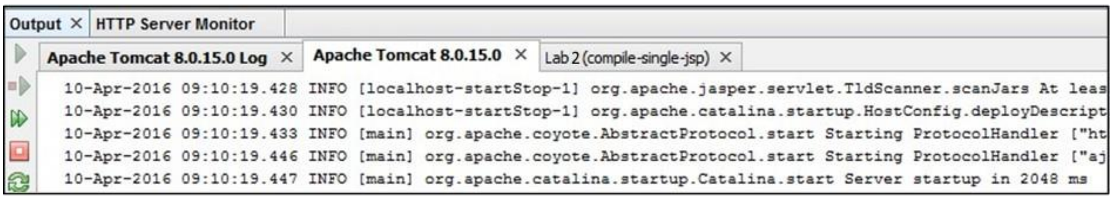
15. Before running any JSP's files for first time upon opening your Netbeans IDE, you need to start your web server (i.e; Apache Tomcat).

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



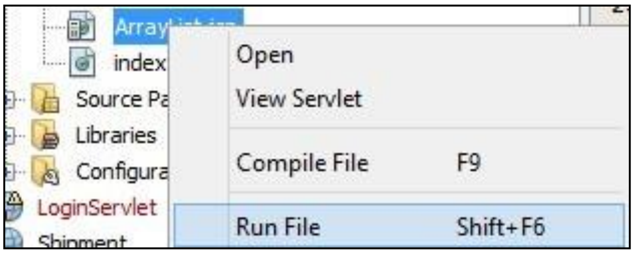
17. You should get the green indicator at *Apache Tomcat*'s icon and *Apache Tomcat* output message with the time taken to start specified time to start *Apache Tomcat* web server.



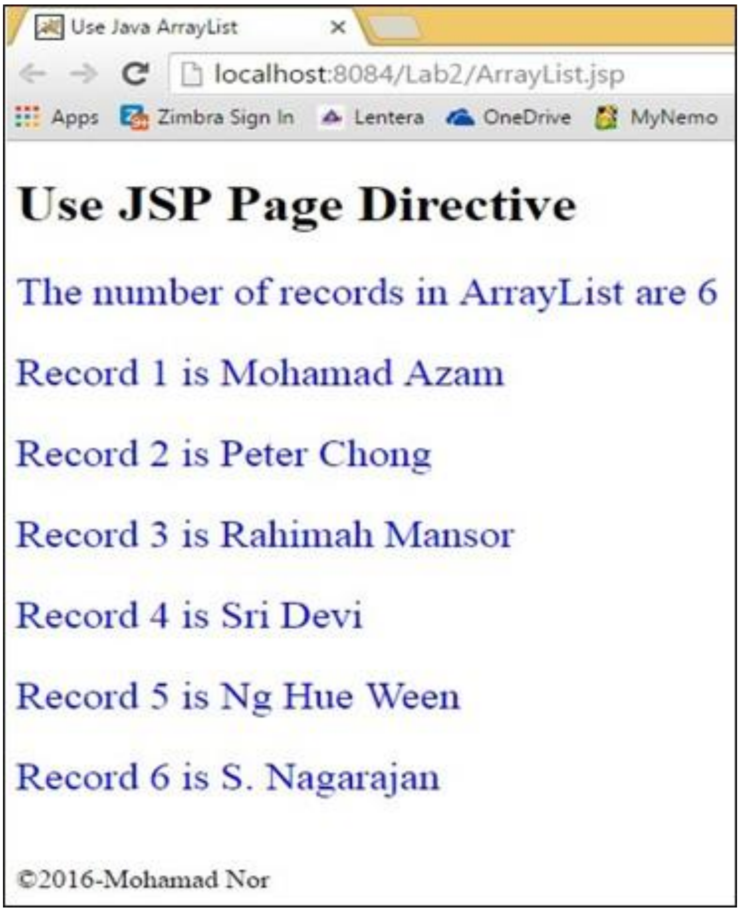


```
10-Apr-2016 09:10:19.428 INFO [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars At leas
10-Apr-2016 09:10:19.430 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDescript
10-Apr-2016 09:10:19.433 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ht
10-Apr-2016 09:10:19.446 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["aj
10-Apr-2016 09:10:19.447 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 2048 ms
```

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.



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?

   Answer:

   Using ArrayList object to store data and retrive it from JSP page.

2. Write a sample syntax how you want to use java *Math* object in JSP?
   Answer:

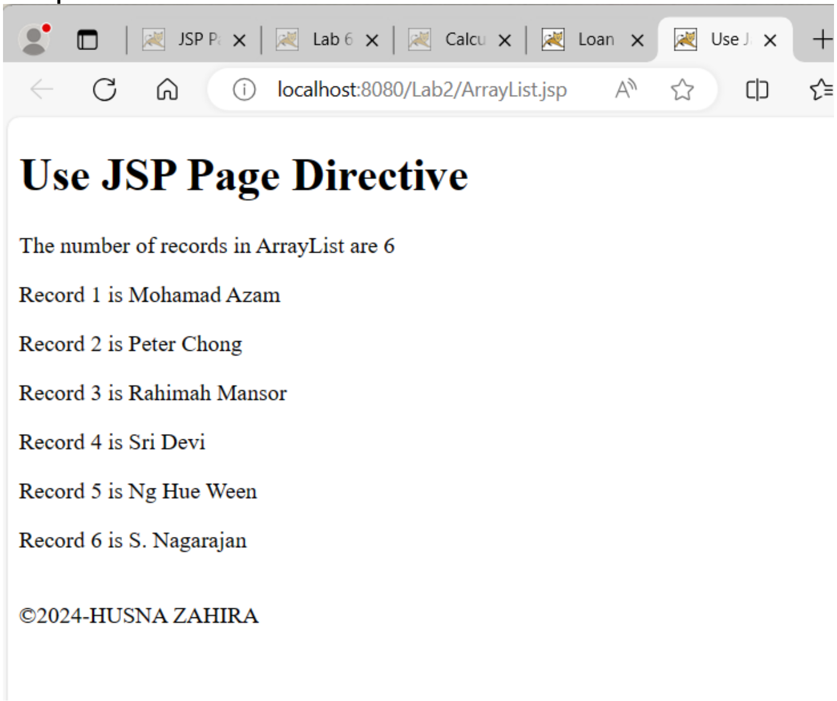
   Double number = 12.7;

   Double squareRoot = Math.sqrt(number);

   Double value = Math.cos(Math.toRadians(45));

3. List and write a sample syntax for THREE (3) of JSP page directive.
   Answer:
   <%@page contentType="text/html" pageEncoding="UTF-8"%>
   <%@page import="java.util.ArrayList"%>
   <%@page import="java.util.List"%>

Output:



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

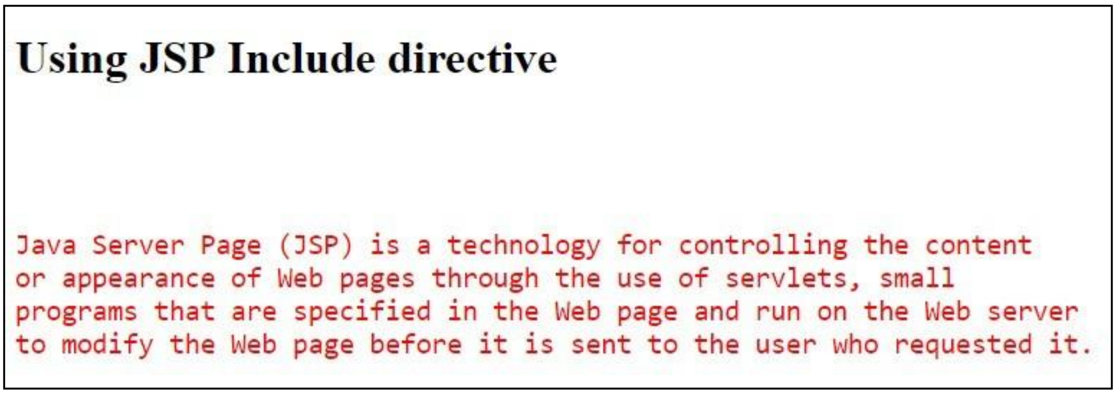
Record 6 is S. Nagarajan

©2024-HUSNA ZAHIRA

Task 6: Use JSP Include directive for JSP Page

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

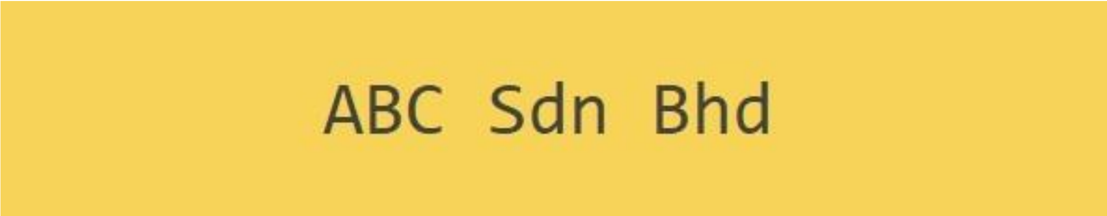
1. Create a new JSP's file.
2. Type file name as *mainPage*.
3. Create content for *mainPage.jsp* as below.



Using JSP Include directive

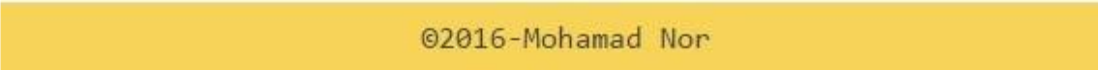
```
Java Server Page (JSP) is a technology for controlling the content
or appearance of Web pages through the use of servlets, small
programs that are specified in the Web page and run on the Web server
to modify the Web page before it is sent to the user who requested it.
```

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



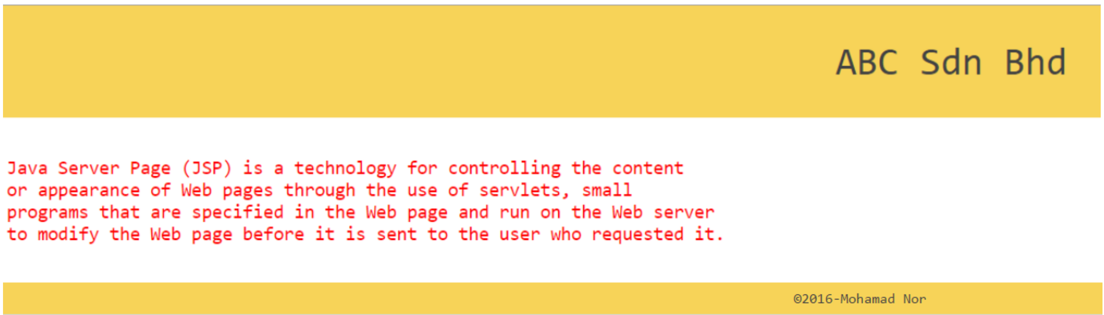
©2016-Mohamad Nor

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 het the following output.



ABC Sdn Bhd

```
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

Reflection

1. What you have learnt from this exercise?

   Answer:

   Learn how to use <%@ include file=”filename.jsp” %> directive into JSP page

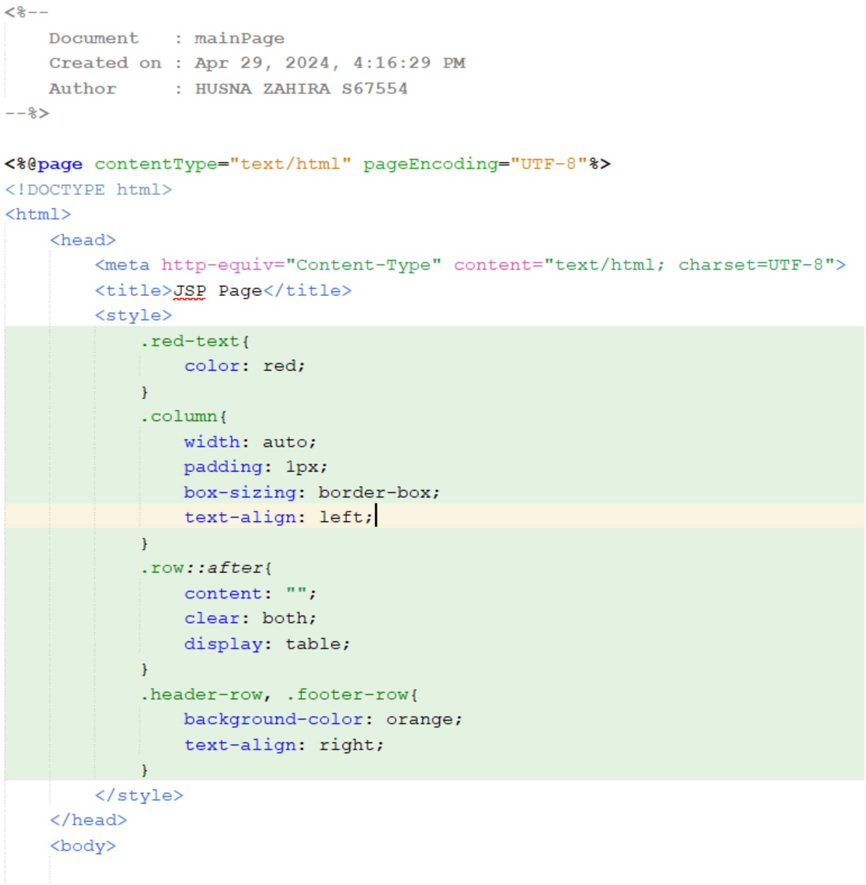
2. Write a syntax how you want to include *common.html* file that located at a directory known as *master*.

   Answer:

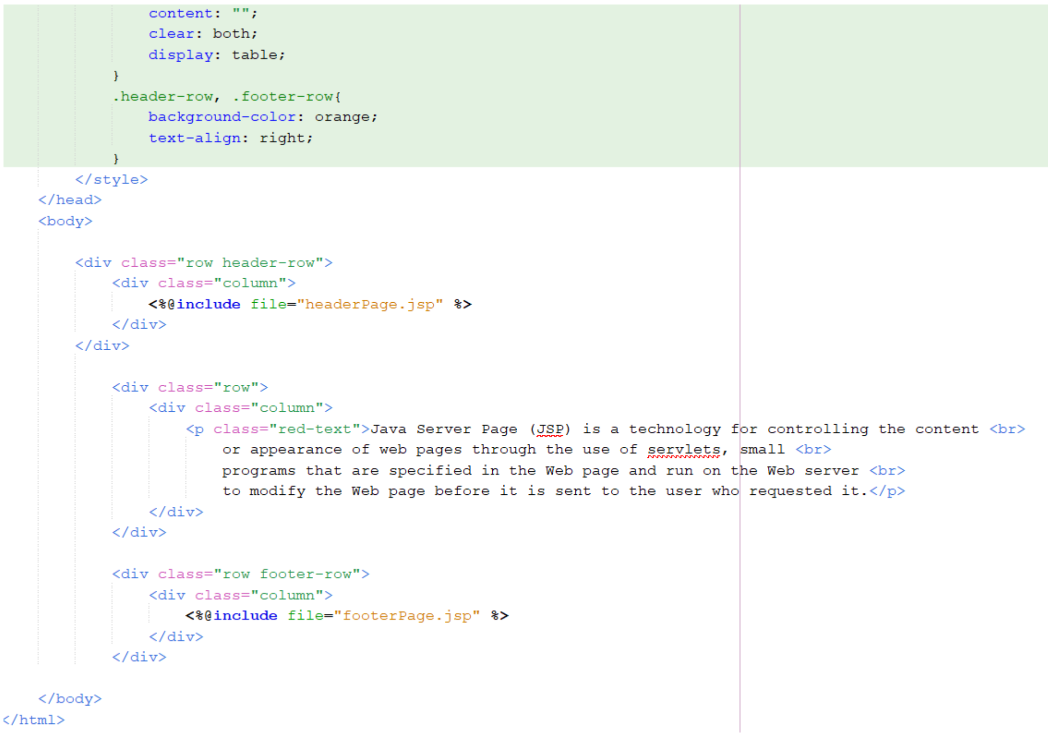
   <%@ include file=”/master/common.html”%>

Code:

mainPage.jsp

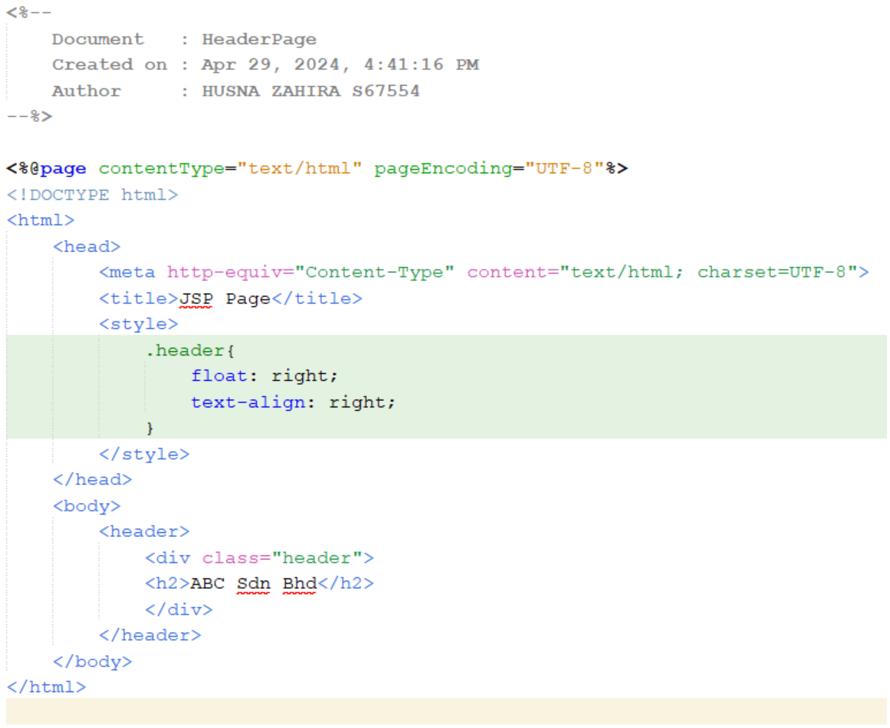


```
<%--
    Document   : mainPage
    Created on : Apr 29, 2024, 4:16:29 PM
    Author     : HUSNA ZAHIRA S67554
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <style>
            .red-text{
                color: red;
            }
            .column{
                width: auto;
                padding: 1px;
                box-sizing: border-box;
                text-align: left;
            }
            .row::after{
                content: "";
                clear: both;
                display: table;
            }
            .header-row, .footer-row{
                background-color: orange;
                text-align: right;
            }
        </style>
    </head>
    <body>
```



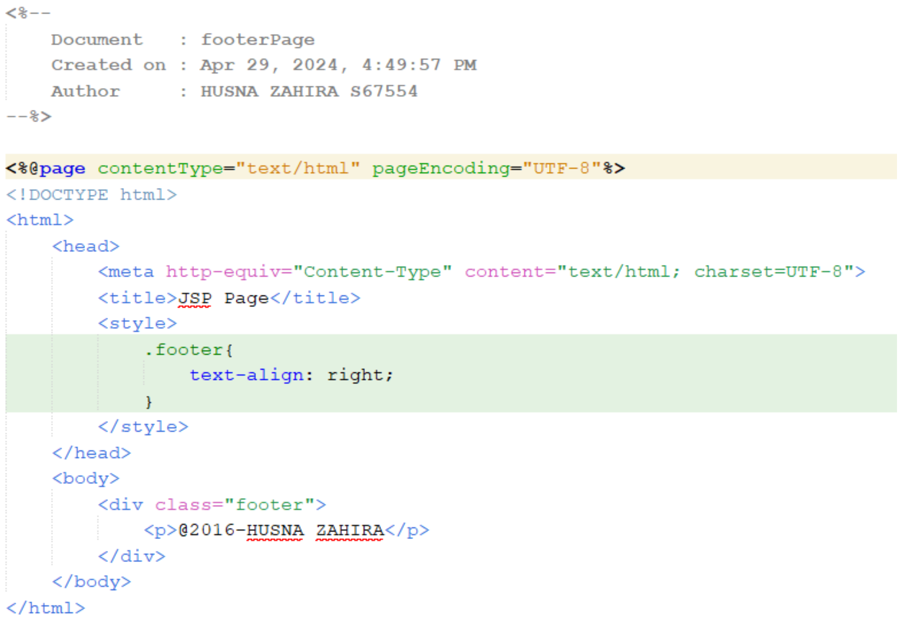
```
                content: "";
                clear: both;
                display: table;
            }
            .header-row, .footer-row{
                background-color: orange;
                text-align: right;
            }
        </style>
    </head>
    <body>

        <div class="row header-row">
            <div class="column">
                <%@include file="headerPage.jsp" %>
            </div>
        </div>

            <div class="row">
                <div class="column">
                    <p class="red-text">Java Server Page (JSP) is a technology for controlling the content <br>
                        or appearance of web pages through the use of servlets, small <br>
                        programs that are specified in the Web page and run on the Web server <br>
                        to modify the Web page before it is sent to the user who requested it.</p>
                </div>
            </div>

            <div class="row footer-row">
                <div class="column">
                    <%@include file="footerPage.jsp" %>
                </div>
            </div>

    </body>
</html>
```

headerPage.jsp



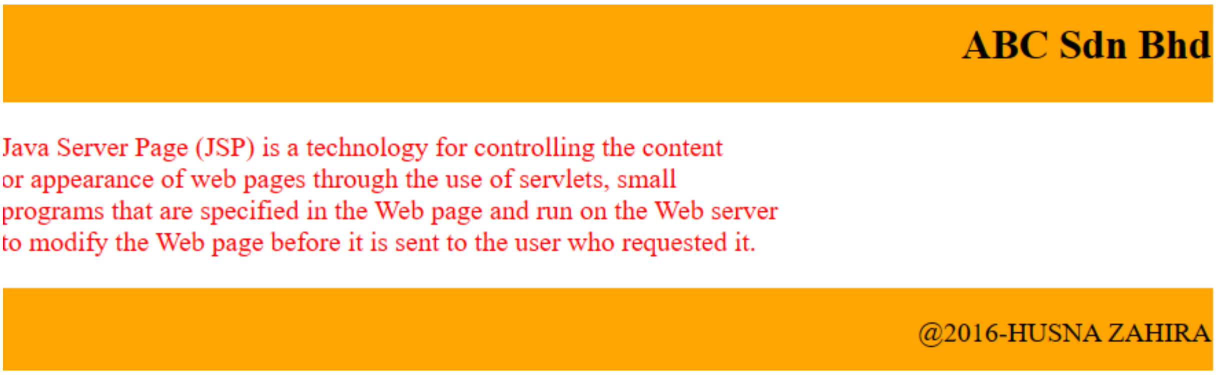
```
<%--
    Document   : HeaderPage
    Created on : Apr 29, 2024, 4:41:16 PM
    Author     : HUSNA ZAHIRA S67554
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <style>
            .header{
                float: right;
                text-align: right;
            }
        </style>
    </head>
    <body>
        <header>
            <div class="header">
            <h2>ABC Sdn Bhd</h2>
            </div>
        </header>
    </body>
</html>
```

footerPage.jsp



```
<%--
    Document   : footerPage
    Created on : Apr 29, 2024, 4:49:57 PM
    Author     : HUSNA ZAHIRA S67554
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <style>
            .footer{
                text-align: right;
            }
        </style>
    </head>
    <body>
        <div class="footer">
            <p>@2016-HUSNA ZAHIRA</p>
        </div>
    </body>
</html>
```

Output:



ABC Sdn Bhd

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-HUSNA ZAHIRA

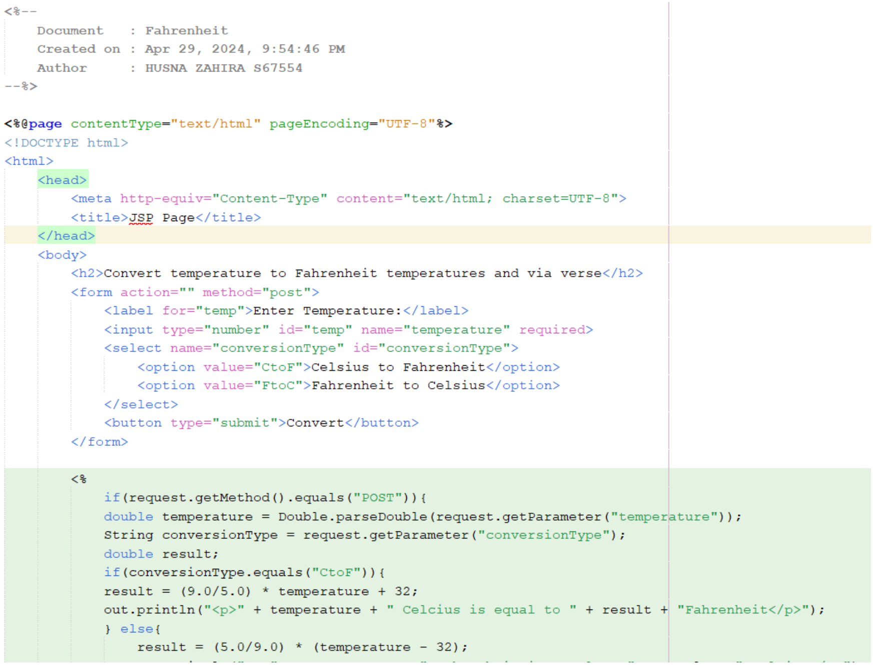
Exercise

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

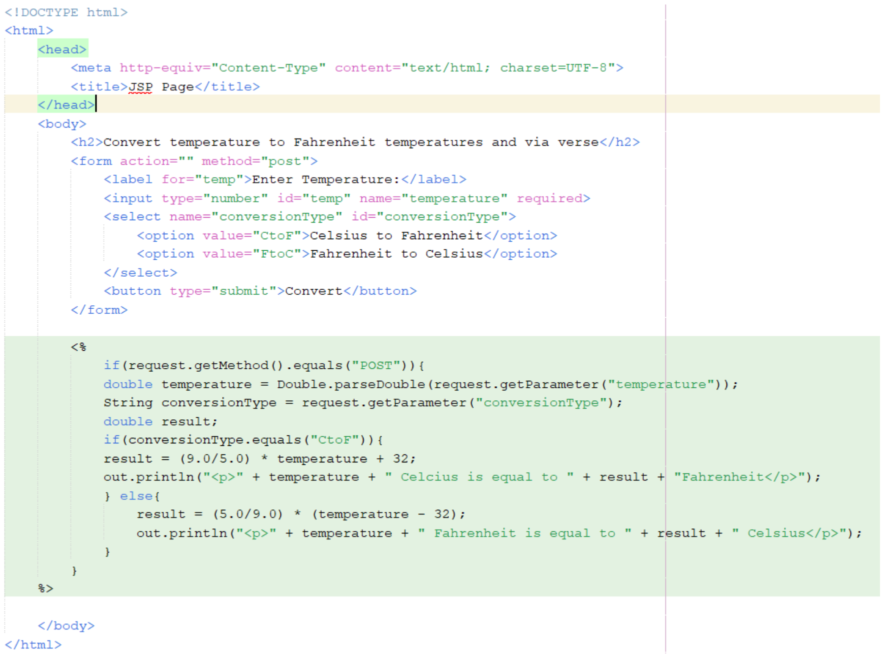
$$F = (9/5)C + 32$$

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

Code:

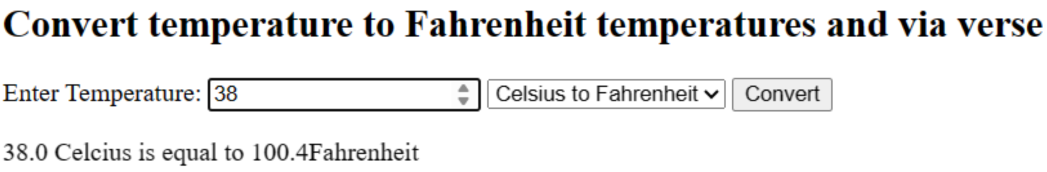


```
<%--
    Document   : Fahrenheit
    Created on : Apr 29, 2024, 9:54:46 PM
    Author     : HUSNA ZAHIRA S67554
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h2>Convert temperature to Fahrenheit temperatures and via verse</h2>
        <form action="" method="post">
            <label for="temp">Enter Temperature:</label>
            <input type="number" id="temp" name="temperature" required>
            <select name="conversionType" id="conversionType">
                <option value="CtoF">Celsius to Fahrenheit</option>
                <option value="FtoC">Fahrenheit to Celsius</option>
            </select>
            <button type="submit">Convert</button>
        </form>

        <%
            if(request.getMethod().equals("POST")){
            double temperature = Double.parseDouble(request.getParameter("temperature"));
            String conversionType = request.getParameter("conversionType");
            double result;
            if(conversionType.equals("CtoF")){
            result = (9.0/5.0) * temperature + 32;
            out.println("<p>" + temperature + " Celcius is equal to " + result + "Fahrenheit</p>");
            } else{
                result = (5.0/9.0) * (temperature - 32);
```



```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h2>Convert temperature to Fahrenheit temperatures and via verse</h2>
        <form action="" method="post">
            <label for="temp">Enter Temperature:</label>
            <input type="number" id="temp" name="temperature" required>
            <select name="conversionType" id="conversionType">
                <option value="CtoF">Celsius to Fahrenheit</option>
                <option value="FtoC">Fahrenheit to Celsius</option>
            </select>
            <button type="submit">Convert</button>
        </form>

        <%
            if(request.getMethod().equals("POST")){
            double temperature = Double.parseDouble(request.getParameter("temperature"));
            String conversionType = request.getParameter("conversionType");
            double result;
            if(conversionType.equals("CtoF")){
            result = (9.0/5.0) * temperature + 32;
            out.println("<p>" + temperature + " Celcius is equal to " + result + "Fahrenheit</p>");
            } else{
                result = (5.0/9.0) * (temperature - 32);
                out.println("<p>" + temperature + " Fahrenheit is equal to " + result + " Celsius</p>");
            }
        }
    %>

    </body>
</html>
```

Output:



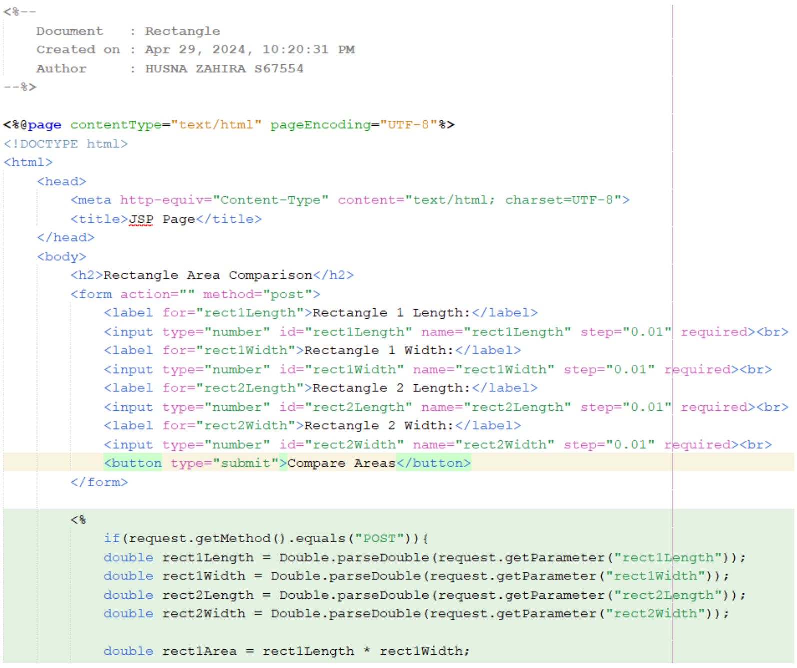
Convert temperature to Fahrenheit temperatures and via verse

Enter Temperature: 38 Celsius to Fahrenheit Convert

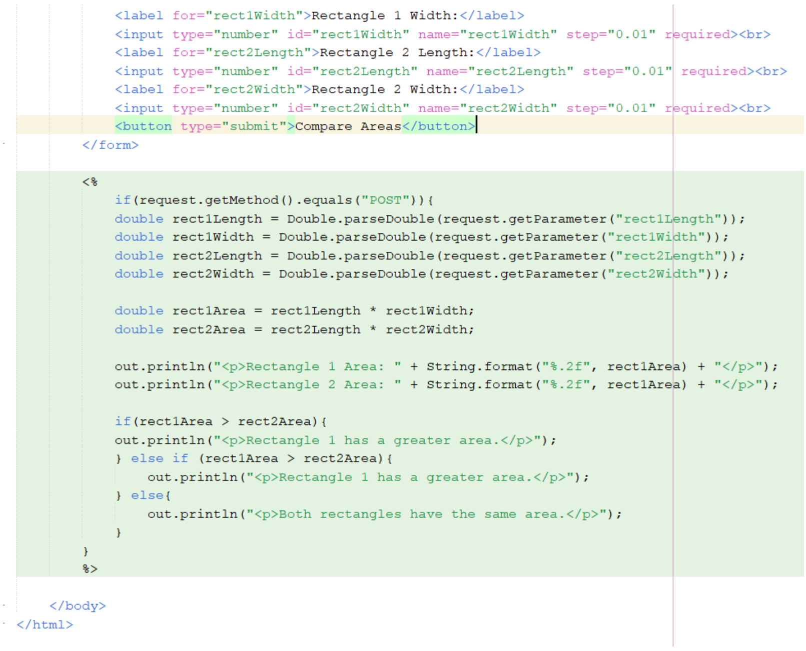
38.0 Celcius is equal to 100.4Fahrenheit

2. 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].

Code:

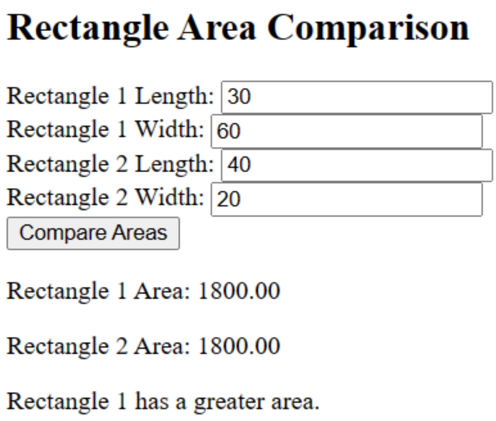


```
<%--
    Document   : Rectangle
    Created on : Apr 29, 2024, 10:20:31 PM
    Author     : HUSNA ZAHIRA S67554
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h2>Rectangle Area Comparison</h2>
        <form action="" method="post">
            <label for="rect1Length">Rectangle 1 Length:</label>
            <input type="number" id="rect1Length" name="rect1Length" step="0.01" required><br>
            <label for="rect1Width">Rectangle 1 Width:</label>
            <input type="number" id="rect1Width" name="rect1Width" step="0.01" required><br>
            <label for="rect2Length">Rectangle 2 Length:</label>
            <input type="number" id="rect2Length" name="rect2Length" step="0.01" required><br>
            <label for="rect2Width">Rectangle 2 Width:</label>
            <input type="number" id="rect2Width" name="rect2Width" step="0.01" required><br>
            <button type="submit">Compare Areas</button>
        </form>

        <%
            if(request.getMethod().equals("POST")){
            double rect1Length = Double.parseDouble(request.getParameter("rect1Length"));
            double rect1Width = Double.parseDouble(request.getParameter("rect1Width"));
            double rect2Length = Double.parseDouble(request.getParameter("rect2Length"));
            double rect2Width = Double.parseDouble(request.getParameter("rect2Width"));

            double rect1Area = rect1Length * rect1Width;
```



```
            <label for="rect1Width">Rectangle 1 Width:</label>
            <input type="number" id="rect1Width" name="rect1Width" step="0.01" required><br>
            <label for="rect2Length">Rectangle 2 Length:</label>
            <input type="number" id="rect2Length" name="rect2Length" step="0.01" required><br>
            <label for="rect2Width">Rectangle 2 Width:</label>
            <input type="number" id="rect2Width" name="rect2Width" step="0.01" required><br>
            <button type="submit">Compare Areas</button>
        </form>

        <%
            if(request.getMethod().equals("POST")){
            double rect1Length = Double.parseDouble(request.getParameter("rect1Length"));
            double rect1Width = Double.parseDouble(request.getParameter("rect1Width"));
            double rect2Length = Double.parseDouble(request.getParameter("rect2Length"));
            double rect2Width = Double.parseDouble(request.getParameter("rect2Width"));

            double rect1Area = rect1Length * rect1Width;
            double rect2Area = rect2Length * rect2Width;

            out.println("<p>Rectangle 1 Area: " + String.format("%.2f", rect1Area) + "</p>");
            out.println("<p>Rectangle 2 Area: " + String.format("%.2f", rect1Area) + "</p>");

            if(rect1Area > rect2Area){
            out.println("<p>Rectangle 1 has a greater area.</p>");
            } else if (rect1Area > rect2Area){
                out.println("<p>Rectangle 1 has a greater area.</p>");
            } else{
                out.println("<p>Both rectangles have the same area.</p>");
            }
        }
        %>

    </body>
</html>
```

Output:



Rectangle Area Comparison

Rectangle 1 Length: 30

Rectangle 1 Width: 60

Rectangle 2 Length: 40

Rectangle 2 Width: 20

Rectangle 1 Area: 1800.00

Rectangle 2 Area: 1800.00

Rectangle 1 has a greater area.