

Week 5

JSP: JavaBeans & Java Standard Tag Library (JSTL)

Web Programming 2

Name: Nur

Arina bt

Abdul

Malek

Matric #:

S65361

Semester:

semester 4

Lab: lab 5

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

Task 1: Using Scriptlet to Access a Simple JavaBeans.....	4
Task 2: Problem Solving using JavaBeans	3
Task 3: Installing JSTL Taglibs	5
Task 4: Using Java Standard Tag Library (JSTL).....	8
Task 5: Using JSP Standard Tag Library.....	14

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.

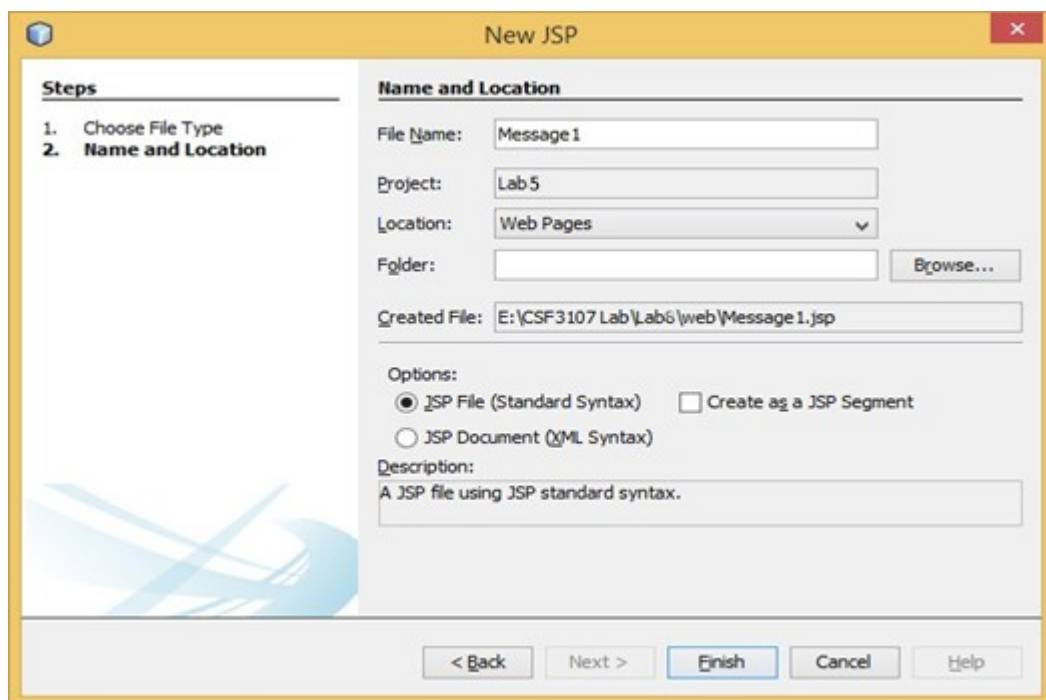
Task 1: Using Scriptlet to Access a Simple JavaBeans

Objective: Use Java Scriptlet to access JavaBeans

Problem Description: Write a JavaBeans that can display message “Welcome to CSF3107 courses....!”.

Estimated time: 20 minutes

1. Go to *Lab5*’s project.
2. Create a new JSP’s file.
3. Type file name as *Message1*.



4. Rename title as *Using JSP Scriptlet*.

5. Rename `<h1>` as *Using JSP Standard Action to call JavaBeans*.

```
9  <html>
10 <head>
11     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12     <title>Using JSP Scriptlet </title>
13 </head>
14 <body>
15     <h1>Using JSP Scriptlet to call JavaBeans</h1>
16 </body>
17 </html>
```

6. Use JSP page directive to include the information such as content type, page info, language, lab8.com package and use java util API.

```
1  <!--
2      Document      : message1
3      Created on    : 18-Apr-2016, 16:13:08
4      Author       : Mohamad Nor Hassan
5  -->
6
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <%@page language="java"%>
9  <%@page info="Using JSP Standard Action to call JavaBeans"%>
10 <%@page import="java.util.Date, lab5.com.Message"%>
```

7. Use a Java scriptlet to create an object for *Message* class.

```
20 <%
21     //Create an object..
22     Message objMsg = new Message();
23
24 %>
```

8. Then, assign the value as “Welcome to CSF3107 course....!” via setter method *setMsg(String msg)*.

```
24     //Assign value..
25     objMsg.setMsg("Welcome to CSF3107 course....!");
```

9. Use *getMsg()*'s method to display the message at paragraph.

```
27     //Display value...
28     out.println("<p>" + objMsg.getMsg() + "</p>");
```

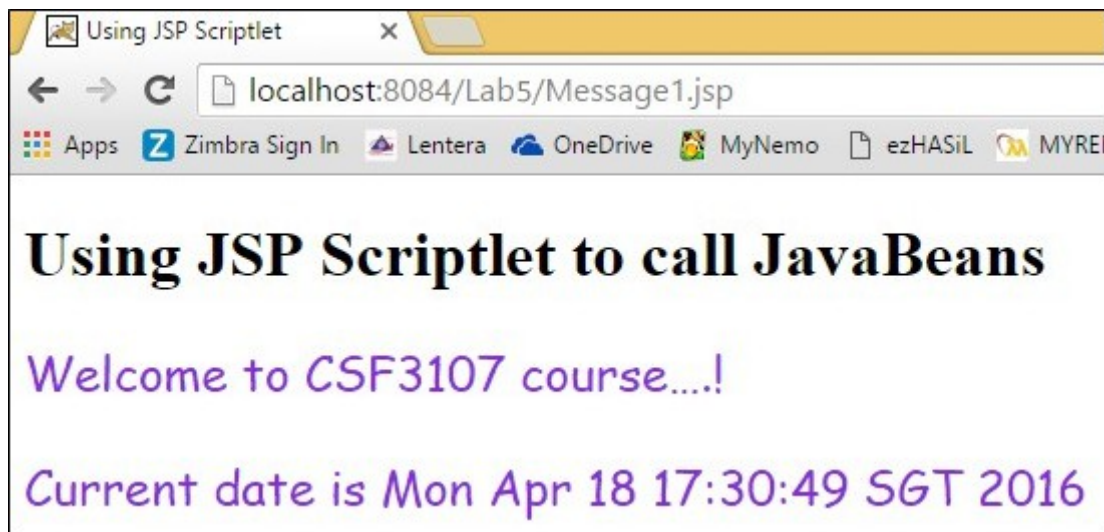
10. Add current date to the next paragraph.

```
30 //Add date..  
31 out.println("<p>Current date is " + new java.util.Date() + "</p>");
```

11. Save *Message1.jsp*

12. Compile and run *Message1.jsp*.

13. You should get the following output.



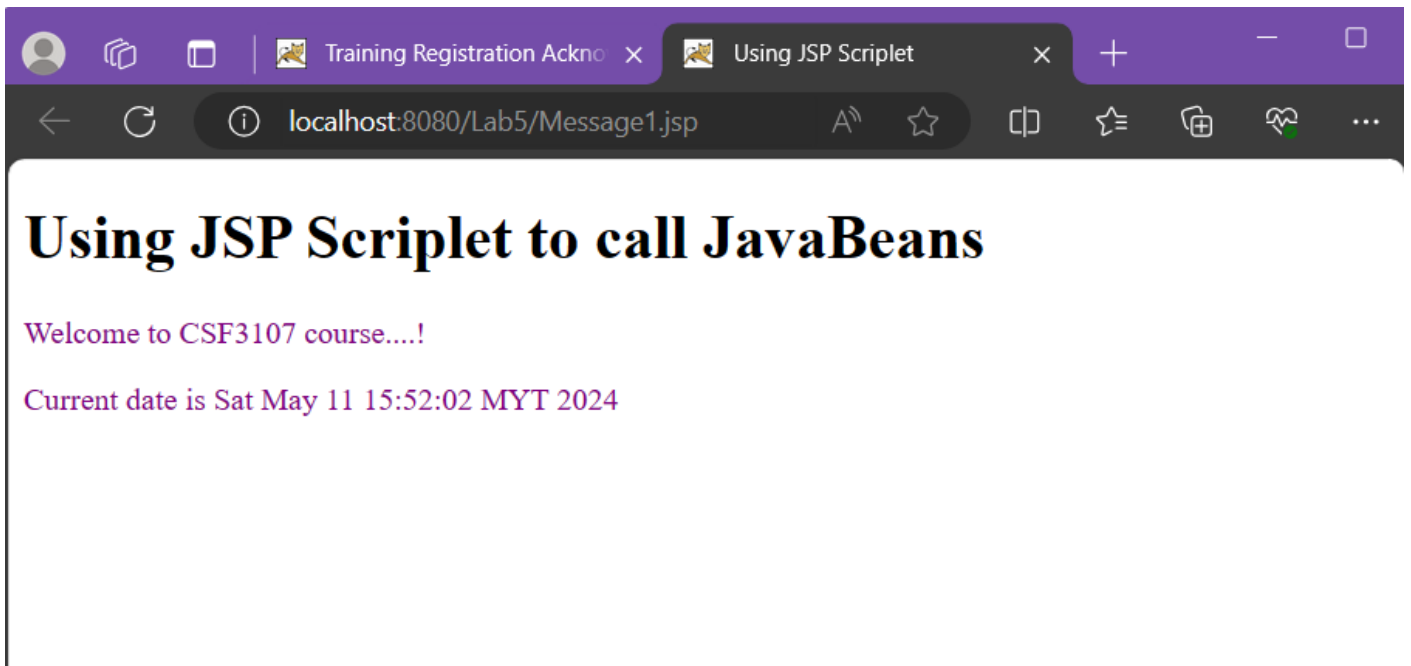
My Work

Message1.jsp

```
7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <%@page language="java"%>
9  <%@page info="Using JSP Standard Action to call JavaBeans"%>
10 <%@page import="java.util.Date, lab5.com.Message"%>
11 <!DOCTYPE html>
12 <html>
13   <head>
14     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
15     <title>Using JSP Scriptlet</title>
16   </head>
17   <body>
18     <h1>Using JSP Scriptlet to call JavaBeans</h1>
19
20     <%
21       //Create an object..
22       Message objMsg = new Message();
23
24       //Assign value..
25       objMsg.setMsg("Welcome to CSF3107 course....!");
26
27       //Display value..
28       out.println("<p style='color: purple;'>" + objMsg.getMsg() + "</p>");
29
30       //Add date..
31       out.println("<p style='color:purple;'>Current date is " + new java.util.Date() + "</p>");
32     %>
33
34   </body>
35 </html>
```

Message.java

```
4
5  package lab5.com;
6
7  /**
8   *
9   * @author ryna
10  */
11  public class Message {
12
13      private String msg;
14
15      public String getMsg() {
16          return msg;
17      }
18
19      public void setMsg(String msg) {
20          this.msg = msg;
21      }
22
23  }
```

Reflection

1. What you have learnt from this exercise?
how to use scriptlet to access javabeans.
2. Explain the differences when calling JavaBeans using JSP Standard Action and Java Scriptlet.

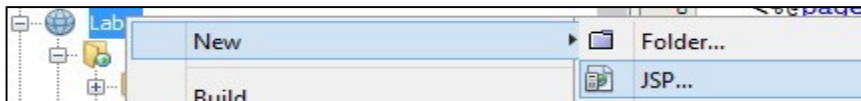
Task 2: Problem Solving using JavaBeans

Objective: Use JavaBeans in JavaScriptlet

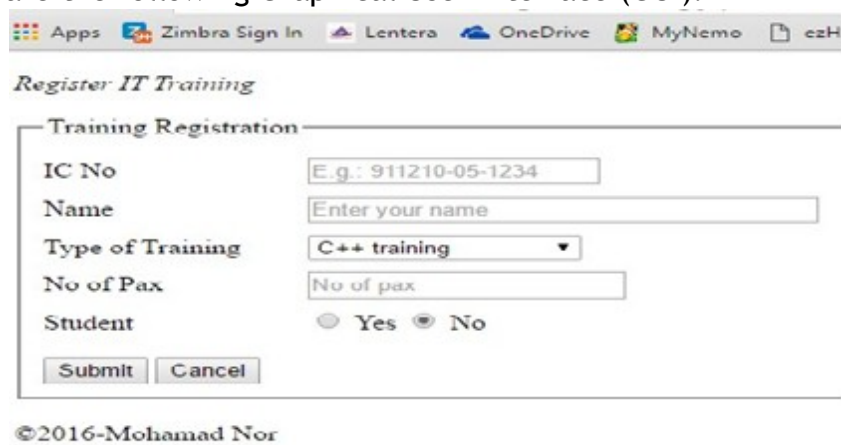
Problem Description: Create sample web form to register IT's training based on the these fees;
C++ training (values as "1")= RM3000 per pax
Java for beginner (values as "2")= RM3000 per pax
HTML5 (values as "3")= RM2800 per pax
Java EEE (values as "4")= RM5500 per pax
Android Programming (values as "5")= RM3200 per pax
Student = Yes (values as "1") and No (Values as "0")
Student will entitle 10% discount
You must cater all front-end validation.

Estimated time: 60 minutes

1. Choose Project *Lab5*.
2. Create a new JSP's file.



3. Type file name as *registerTraining*.
4. Prepare the following Graphical User Interface (GUI).

A screenshot of a web browser displaying a form titled 'Register IT Training'. The form has the following fields: 'IC No' with a placeholder 'E.g.: 911210-05-1234', 'Name' with a placeholder 'Enter your name', 'Type of Training' with a dropdown menu showing 'C++ training', 'No of Pax' with a placeholder 'No of pax', and 'Student' with radio buttons for 'Yes' and 'No'. At the bottom are 'Submit' and 'Cancel' buttons. The footer of the page reads '©2016-Mohamad Nor'.

5. Create a *Register* JavaBeans to cater the business requirements and store it into package *lab8.com*.
6. Create a new file name known as *processTraining.jsp*.

7. Attached the *pocessTraining.jsp*'s form to *registerTraining.jsp*.
8. Open *pocessTraining.jsp*'s and use Java Scriptlet to invoke *Register* JavaBeans.
9. Output will appear in web browser.



← → ↻ localhost:8084/Lab8/processTraining.jsp

Apps Zimbra Sign In Lentera OneDrive MyNemo ezHASiL MYREN Clo

Training Registration Acknowledgement

IC No : 910710-11-2416

Name : Mohamad Nor Hassan

Type of Training : Java EEE

Number of Pax : 1 person/s

Student : Yes

Amount Due : RM 4950.00

©2016-Mohamad Nor

The image shows two screenshots of a web browser. The top screenshot displays the 'Register IT Training' page, which contains a registration form. The bottom screenshot displays the 'Training Registration Acknowledgement' page, which shows the details of the registration.

Register IT Training

Training Registration

Ic No * 910710-11-2416

Name Mohamad Nor Hassan

Type of Training Java EEE

No of Pax 1 person/s

Student ☒ Yes ☐ No

Submit Cancel

Training Registration Acknowledgement

IC No: 910710-11-2416

Name: Mohamad Nor Hassan

Type of Training: JavaTraining

No of Pax: 0

Student Status: yes

Amount Due: RM 4950.00

registerTraining.jsp

```

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>Register Training</title>
13 </head>
14 <body>
15 <h1>Register IT Training</h1>
16 <form id="registerFrm" action="processTraining.jsp" method="post" onsubmit="return checkICNo()">
17 <fieldset>
18 <legend>Training Registration</legend>
19 <label for="invoiceno">IC No *</label>
20 <input type="text" id="icno" name="my_icno" size="15" placeholder="E.g. 911210-05-1234"><br/>
21
22 <label for="name">Name</label>
23 <input type="text" id="name" name="my_name" size="45" placeholder="Enter your name"><br/>
24
25 <label for="Type of Training">Type of Training</label>
26 <select id="Type of Training" name="Training">
27 <option value="JavaTraining">Java <del>EEE</del></option>
28 <option value="C++Training">C++ Training</option>
29 <option value="PythonTraining">Python Training</option>
30 </select><br>
31
32 <label for="noOfPax">No of Pax</label>
33 <input type="text" id="noOfPax" name="my_noofpax" size="45" placeholder="No of Pax"><br/>
34
35 <label for="yes">Student</label>
36 <input type="radio" id="yes" name="student" value="yes"><label for="yes">Yes</label>
37 <input type="radio" id="no" name="student" value="no"><label for="no">No</label>
38
39
40 <p><input type="submit" id="btnSubmit" value="Submit"/>
41 <input type="reset" id="btnCancel" value="Cancel"/>
42 </p>
43 </form>
44 </body>
45 </html>
46

```

processTraining.jsp

```

1  <%@ page import="lab8.com.Register" %>
2  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
3  <%! Register register = new Register(); %>
4  <!DOCTYPE html>
5  <html>
6  <head>
7      <meta charset="UTF-8">
8      <title>Training Registration Acknowledgement</title>
9      <style>
10         .purple-text {
11             color: purple;
12         }
13     </style>
14 </head>
15 <body>
16     <h1>Training Registration Acknowledgement</h1>
17     <%
18         register.setIcNo(request.getParameter("my_icno"));
19         register.setName(request.getParameter("my_name"));
20         register.setTrainingType(request.getParameter("Training"));
21         String noOfPaxStr = request.getParameter("my_noofpax");
22         register.setStudentStatus(request.getParameter("student"));
23     %>
24     <p class="purple-text">IC No: <%= register.getIcNo() %></p>
25     <p class="purple-text">Name: <%= register.getName() %></p>
26     <p class="purple-text">Type of Training: <%= register.getTrainingType() %></p>
27     <p class="purple-text">No of Pax: <%= register.getNoOfPax() %></p>
28     <p class="purple-text">Student Status: <%= register.getStudentStatus() %></p>
29     <p class="purple-text">Amount Due: RM 4950.00</p>
30 </body>
31 </html>

```

Register.java

```
package lab8.com;

import java.io.Serializable;

public class Register implements Serializable {
    private String icNo;
    private String name;
    private String typeOfTraining;
    private int noOfPax;
    private String studentStatus;

    public String getIcNo() {
        return icNo;
    }

    public void setIcNo(String icNo) {
        this.icNo = icNo;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String get.TypeOfTraining() {
        return typeOfTraining;
    }
}
```

```
31
32     public String getTypeOfTraining() {
33         return typeOfTraining;
34     }
35
36     public void setTypeOfTraining(String typeOfTraining) {
37         this.typeOfTraining = typeOfTraining;
38     }
39
40     public int getNoOfPax() {
41         return noOfPax;
42     }
43
44     public void setNoOfPax(int noOfPax) {
45         this.noOfPax = noOfPax;
46     }
47
48     public String getStudentStatus() {
49         return studentStatus;
50     }
51
52     public void setStudentStatus(String studentStatus) {
53         this.studentStatus = studentStatus;
54     }
55 }
56
```


Reflection

1. What you have learnt from this exercise?
2. Describe the steps how you construct *Register* JavaBeans?

Task 3: Installing JSTL Taglibs

Objective: Installation of Apache Taglibs

Problem Description: To install JSTL Library

Estimated time: 25 minutes

1. Go to the browser and type URL

<http://tomcat.apache.org/taglibs/index.html>

Apache Taglibs

This project is an open source repository for JSP(tm) Tag Libraries.

In particular, Apache Taglibs hosts the [Apache Standard Taglib](#), an implementation of the JSTL specification. Various versions are available.

2. Click on *Apache Standard Taglib*.

Standard Taglib

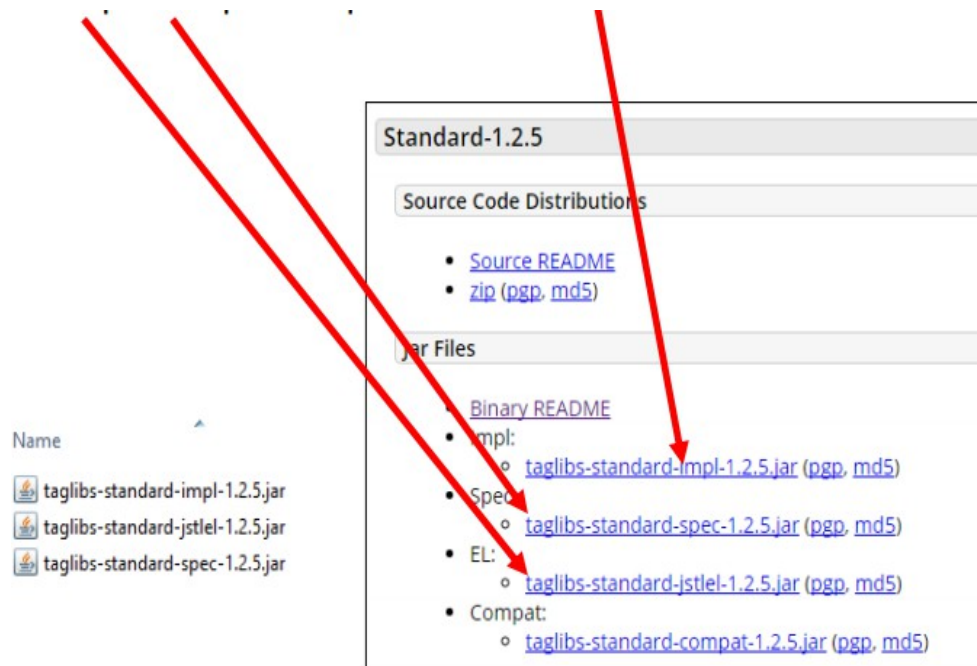
JSP(tm) Standard Tag Library implementations

Apache hosts the Apache Standard Taglib, an implementation of the [JSP Standard Tag Library \(JSTL\)](#) specification. Various versions are available.

Version	JSTL version	Requirements	Getting the Taglib
Standard 1.2.3	JSTL 1.2	Servlet 2.5, JavaServer Pages 2.1	download (javadoc)
Standard 1.1	JSTL 1.1	Servlet 2.4, JavaServer Pages 2.0	download (javadoc)
Standard 1.0	JSTL 1.0	Servlet 2.3, JavaServer Pages 1.2	download (javadoc)

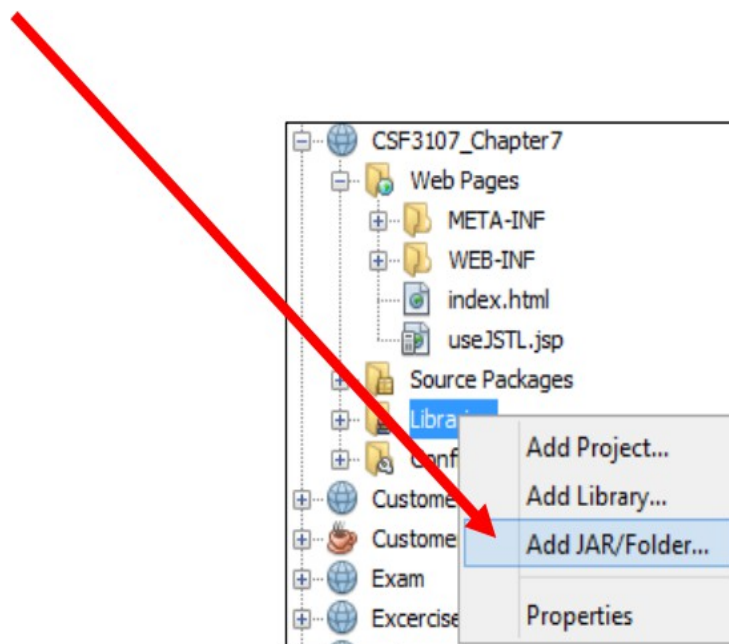
3. Choose Version 1.2 and click download link.
4. Click to jar files for Impl and save the link.

5. Repeat step 3 for Spec and EL.

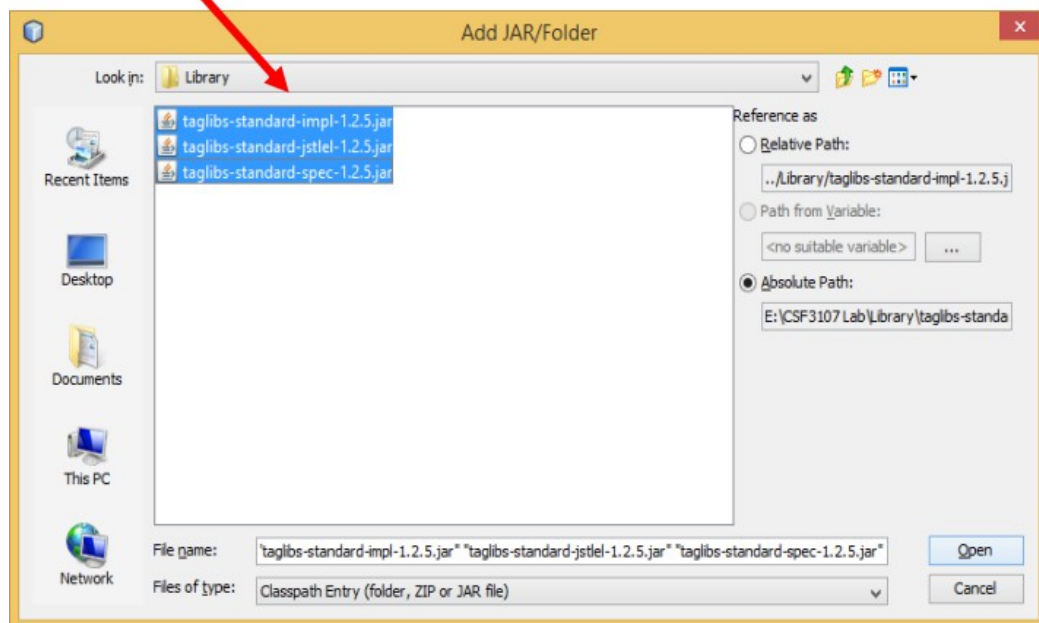


6. Go to your project -> Libraries.

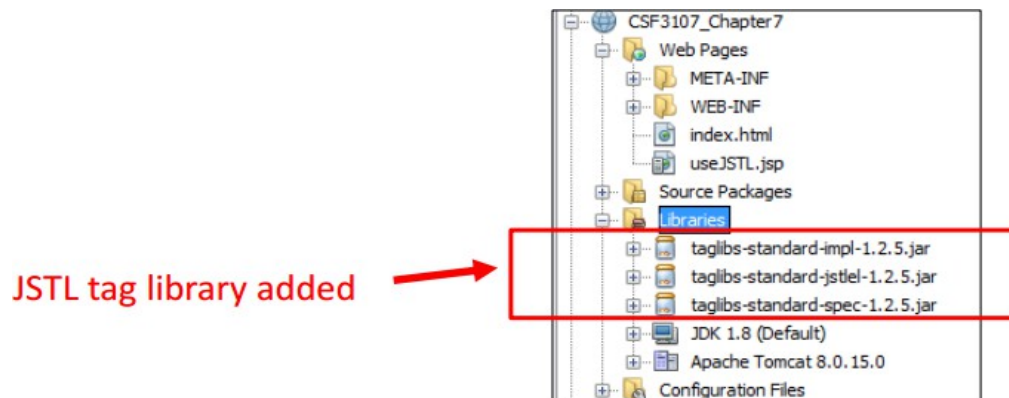
7. Right click your mouse and click to Add JAR/Folder.



8. Choose taglibs-standard-impl-1.2.5.jar, taglibs-standard-jstlel-1.2.5.jar and taglibs-standard-spec-1.2.5.jar and click Open button.



9. You will see all jars files; taglibs-standard-impl-1.2.5.jar, taglibs-standardjstlel-1.2.5.jar and taglibs-standard-spec-1.2.5.jar successfully added in Libraries's folder in NetBeans.



Reflection

What you have learnt from this exercise?

Task 4: Using Java Standard Tag Library (JSTL)

Objective: Using JSTL core tags directive to retrieve information from one page and display it in another JSP page and format number, currency.

Problem Description:

1. To set the value “Welcome to CSF3107 - Web Programming courses..!” and display the value using JSTL core.
2. To passing information from one page to another using JSTL core
 - i. Create userRegistration.html to key-in information.
 - ii. Create userHandler.jsp to receive information key-in from userRegistration.html.
3. Using JSTL’s fmt to format number and currency.

Estimated time: 30 minutes

Prerequisite: Must attached JSTL’s library to current Project’s directory (Refer to Chapter 7’s notes).

Task 1 Assign value “ Welcome to CS F3107-Web Programming 2 courses” and display the value inside J SP page

1. Go to project *Lab5*.
2. Create a new JSP’s file as *jstlCore1*.

3. Rename the title as *Using JSTL tag library* and `<h1>` as *Use JSTL's features*.

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Using JSTL tag library</title>
  </head>
  <body>
    <h1>Use JSTL's features</h1>
  </body>
</html>
```

4. Before using JSTL's tag, we need to declare JSTL's library tag using Taglib directive.

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

5. Using JSTL core tag, assign the value "Welcome to CSF3107 - Web Programming courses..!" to specific variable and finally, display the message.

```
<body>
  <h1>Use JSTL's features</h1>
  <c:set var="message" value="Welcome to CSF3107 - Web Programming courses..!" />
  <p> <c:out value="${message}" /> </p>
</body>
```

- Before you compile *jstlCore1.jsp*, please ensure you already attach jar file for JSTL tag library.



- Compile and run *jstlCore1.jsp* page.

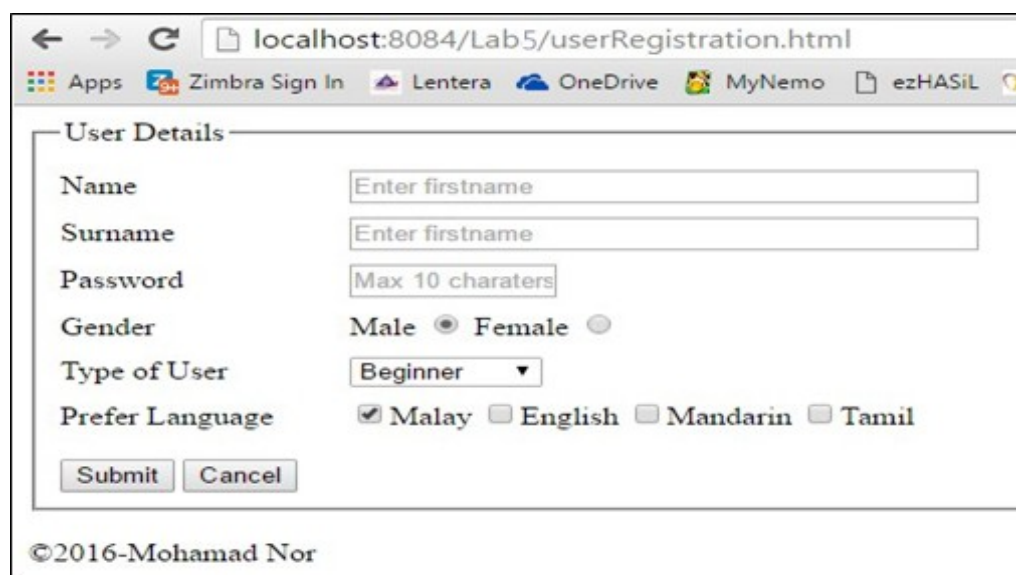
- You will get the following output.



Task 2-Using JSTL's core to request information from one page and display the information

Step 1

- Go to Lab8's project.
- Create HTML's file and rename as userRegistration.
- Produce the following HTML's form.



localhost:8084/Lab5/userRegistration.html

Apps Zimbra Sign In Lentera OneDrive MyNemo ezHASiL MYREN Cloud

User Details

Name

Surname

Password

Gender Male ☒ Female ☐

Type of User

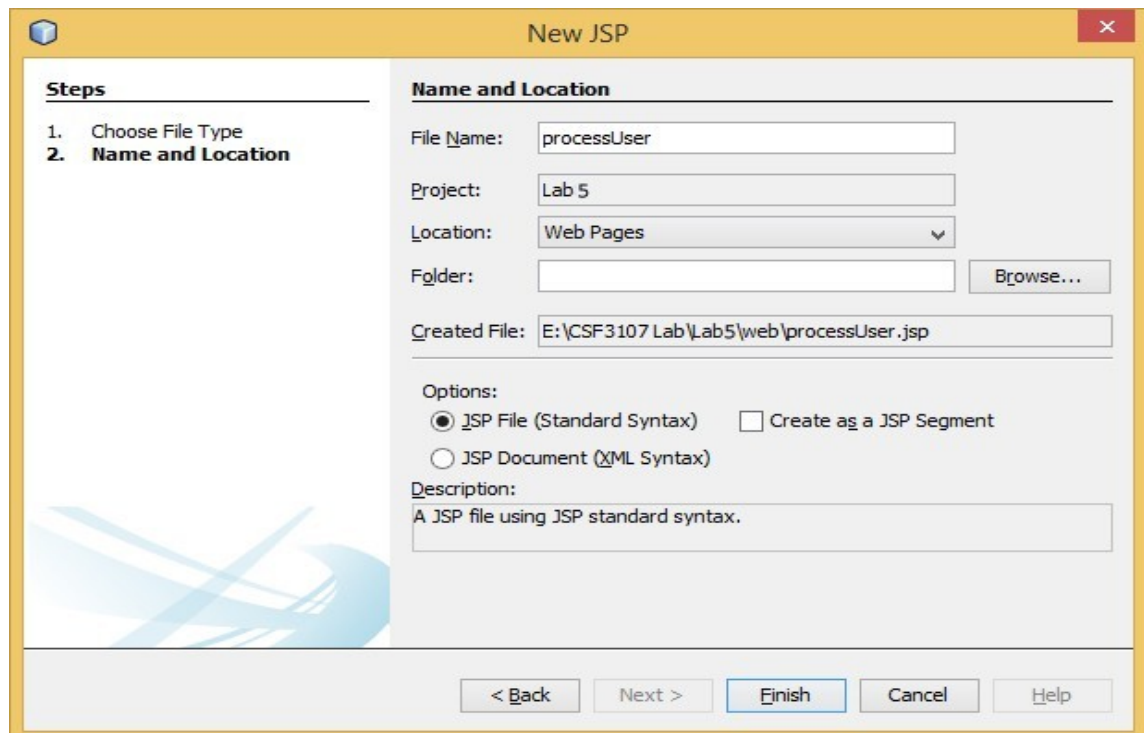
Prefer Language ☒ Malay ☐ English ☐ Mandarin ☐ Tamil

©2016-Mohamad Nor

Note: Types of user is *Beginner*, *Intermediate* and *Advanced*

Step 2

1. Create JSP's file and rename as *processUser*.



2. Add JSTL taglib directive into the *processUser.jsp*'s page.
3. Retrieve the information by using *c:param* tag and display the information by using *c:out* tag.
4. Compile *userHandler.jsp* page.
5. Run *userRegistration.html* page and key-in the information.

User Details

Name: Mohamad Nor

Surname: Hassan

Password:

Gender: Male ☒ Female ☐

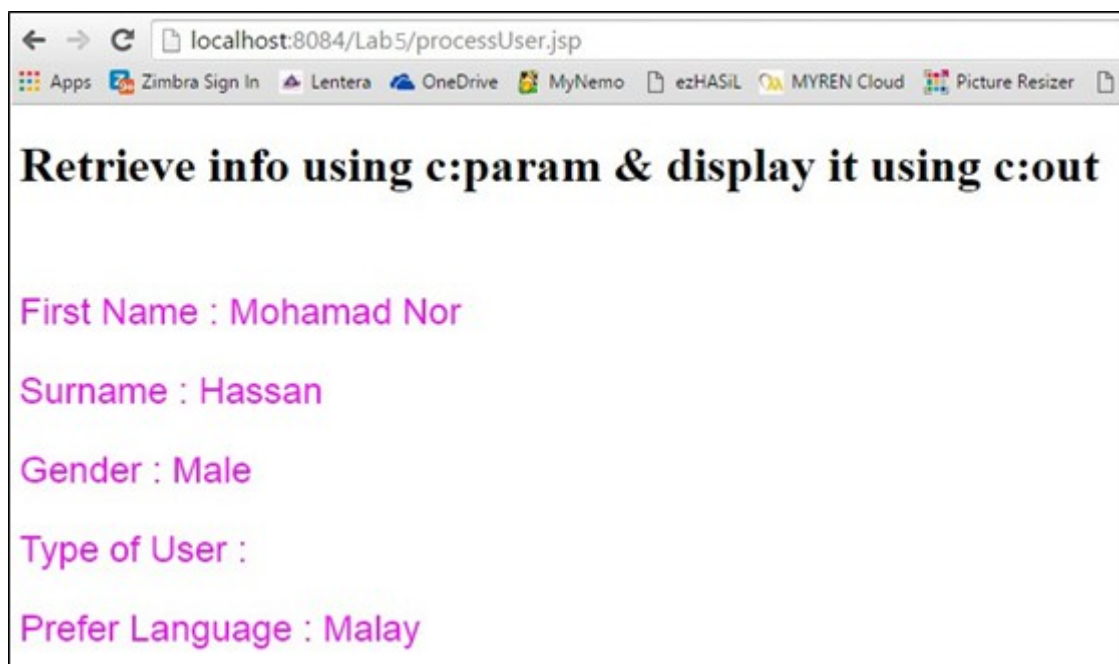
Type of User: Advanced

Prefer Language: ☒ Malay ☐ English ☐ Mandarin ☐ Tamil

©2016-Mohamad Nor

6. Click *Submit* button.

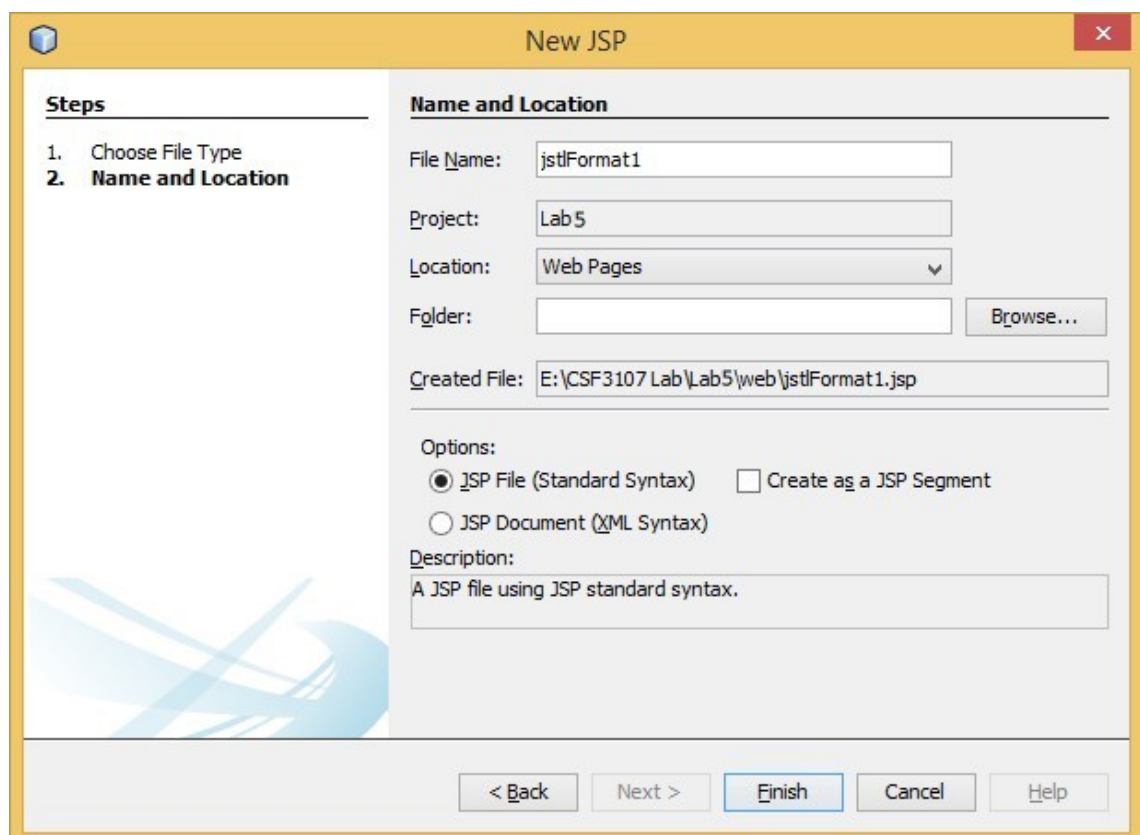
7. You will get the following output.



Task 3 - Using JSTL's fmt to format number and currency

Step 1

1. Create JSP's file and rename as *jstlFormat1*.



2. Add JSTL taglib directive for JSTL *core* and JSTL *formatting* into the current code.
3. Add the following code in your page.

```
<body>
  <h1>Using JSTL formatting tag for formatting</h1>

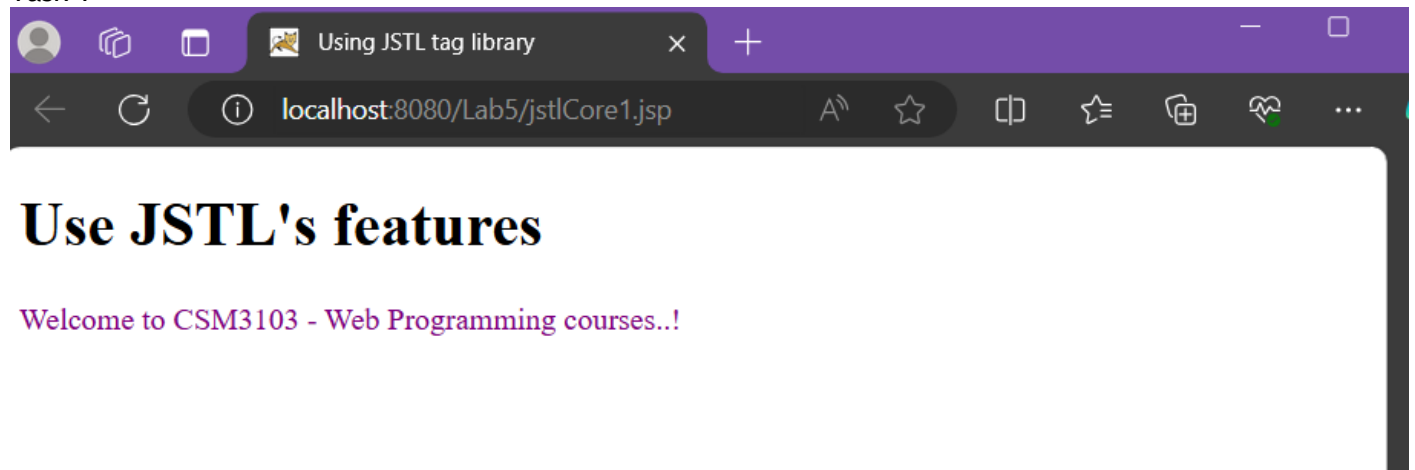
  <!-- Assign specific number to variable -->
  <c:set var="total" value="2880.4638"/>
  <p>Number to be formatted is <c:out value="${total}" /></p>
  <p>Formatting number as currency with currency code : <fmt:formatNumber type="currency" currencyCode="MYR" value="${total}" /></p>
  <p>Formatting number to the nearest 2 integer digit : <fmt:formatNumber type="number" maxIntegerDigits="2" value="${total}" /></p>
  <p>Formatting number by grouping : <fmt:formatNumber type="number" groupingUsed="true" value="${total}" /></p>
</body>
```

4. Continue to do formatting for
 - 3 decimal places
 - percentage
5. Save *jstlFormat1.jsp*
6. Compile and run *jstlFormat1.jsp*.
7. You will get the following output.



My Work

Task 1



```
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
9 <%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14 <title>JSP Page</title>
15 <style>
16 <pre>
17     p {
18         color : purple;
19     }
20 </pre>
21 </style>
22 </head>
23 <body>
24 <h1>Using JSTL formatting tag for formatting</h1>
25
26 <c:set var="total" value="2880.4638"/>
27 <p>Number to be formatted is <c:out value="${total}"/></p>
28 <p>Formatting number as currency with currency code : <fmt:formatNumber type="currency" currencyCode="MYR" value="${total}"/></p>
29 <p>Formatting number to the nearest 2 digit : <fmt:formatNumber type="number" maxIntegerDigits="2" value="${total}"/></p>
30 <p>Formatting number by grouping : <fmt:formatNumber type="number" groupingUsed="true" value="${total}"/></p>
31
32 <p>Formatting number by 3 decimal places : <fmt:formatNumber type="number" maxFractionDigits="3" value="${total}"/></p>
33 <p>Formatting number by % : <fmt:formatNumber type="percent" value="${total}"/></p>
34 </body>
35 </html>
```

Task 2

userRegistration.html

```
<html>
  <head>
    <title>User Details</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>

    <form id="userRegistration" action="processUser.jsp" method="post">
      <fieldset><legend>User Details</legend>
      <label for="name">Name</label>
      <input type="text" id="name" name="name" required><br><br>

      <label for="surname">Surname</label>
      <input type="text" id="surname" name="surname" required><br><br>

      <label for="password">Password</label>
      <input type="password" id="password" name="password" required><br><br>

      <label for="gender">Gender</label>
      <input type="radio" id="male" name="gender" value="male">
      <label for="male">Male</label>
      <input type="radio" id="female" name="gender" value="female">
      <label for="female">Female</label><br><br>

      <label for="typeuser">Type of User</label>
      <select name="typeuser" id="typeuser">
        <option value="Beginner">Beginner</option>
        <option value="Intermediate">Intermediate</option>
        <option value="Advance">Advance</option>
      </select><br><br>

      </select><br><br>

      <label for="preflang">Preferred Language</label><br>
      <input type="checkbox" id="malay" name="preflang" value="malay">
      <label for="malay"> Malay </label>
      <input type="checkbox" id="english" name="preflang" value="english">
      <label for="english"> English </label>
      <input type="checkbox" id="mandarin" name="preflang" value="mandarin">
      <label for="mandarin"> Mandarin </label>
      <input type="checkbox" id="tamil" name="preflang" value="tamil">
      <label for="tamil"> Tamil </label><br><br>

      <input type="submit" id="btnSubmit" value="Submit">
      <input type="reset" id="btnCancel" value="Cancel">
    </fieldset>
  </form>
</body>
</html>
```

Using JSTL tag library

User Details

localhost:8080/Lab5/userRegistrat...

User Details

Name

Mohamad Nor

Surname

Hassan

Password

.....

Gender

☒ Male
☐ Female

Type of User

Advance

Preferred Language

☒ Malay
☐ English
☐ Mandarin
☐ Tamil

Submit

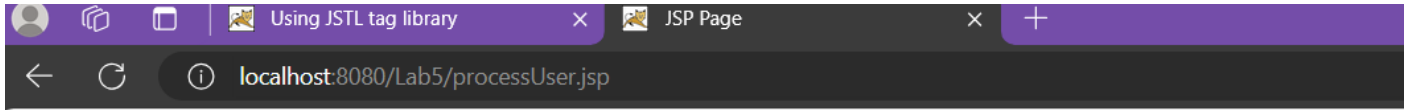
Cancel

processUser.jsp

```

7  <%@page contentType="text/html" pageEncoding="UTF-8"%>
8  <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
9  <!DOCTYPE html>
10 <html>
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13 <title>JSP Page</title>
14 <style>
15     p {
16         color : purple;
17     }
18 </style>
19 </head>
20 <body>
21 <c:url var="userInfoUrl" value="userHandler.jsp">
22 <c:param name="name" value="${param.name}" />
23 <c:param name="surname" value="${param.surname}" />
24 <c:param name="password" value="${param.password}" />
25 <c:param name="gender" value="${param.gender}" />
26 <c:param name="typeuser" value="${param.typeuser}" />
27 <c:param name="preflang" value="${param.preflang}" />
28 </c:url>
29
30 <h1>Retrieve info using c:param & display it using c:out</h1>
31 <p>First Name: <c:out value="${param.name}" /></p>
32 <p>Surname: <c:out value="${param.surname}" /></p>
33 <p>Gender: <c:out value="${param.gender}" /></p>
34 <p>Type of User: <c:out value="${param.typeuser}" /></p>
35 <p>Preferred Languages: <c:out value="${param.preflang}" /></p>
36
37 </body>

```



Retrieve info using c:param & display it using c:out

First Name: Mohamad Nor

Surname: Hassan

Gender: male

Type of User: Advance

Preferred Languages: malay

Task 3

Reflection

1. What the purpose of using JSTL's tag library?

Encapsulation of Common Tasks: JSTL tags encapsulate common tasks such as iteration, conditional processing, formatting, XML processing, and database access, reducing the need for embedded Java code (scriptlets) in JSP pages.

2. List **FIVE(5)** categories of JSTL library.

- Core tags
- Formatting tags
- XML tags
- SQL tags
- Functions tags

Task 5: Using JSP Standard Tag Library

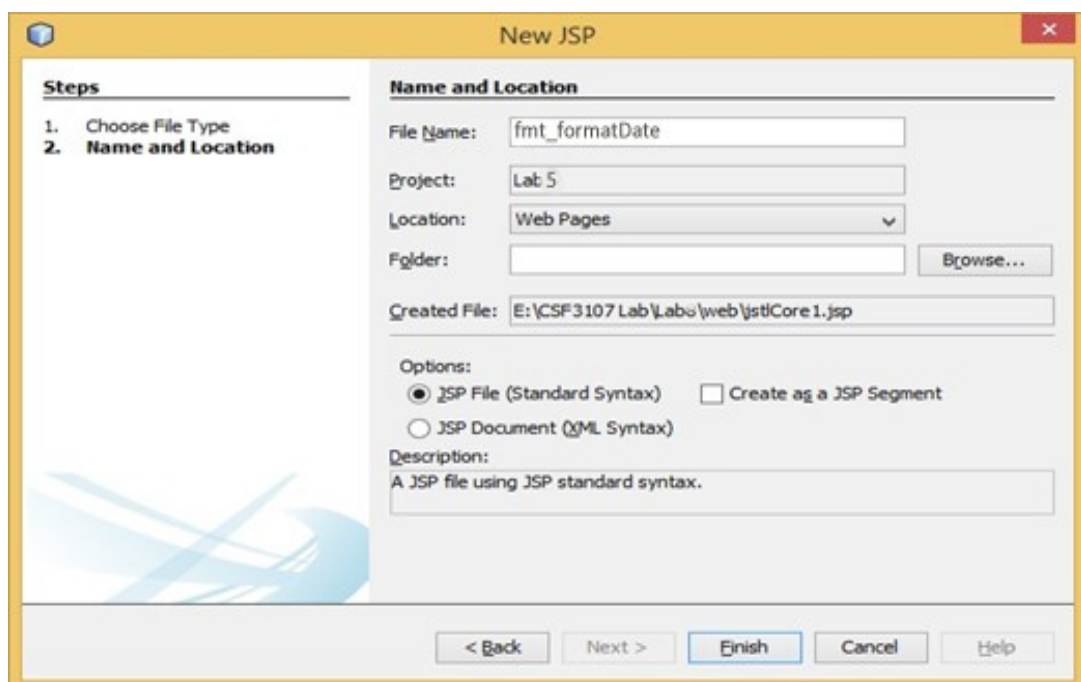
Objective: Using JSTL tags for parsing and formatting the dates.

Problem Description: 1. Using JSTL's `fmt` to format Dates.
2. Using JSTL's `fmt` to parse Dates.

Estimated time: 30 minutes

Task 1 Using JSTL's `fmt` to format Date

1. Go to project *Lab5*.
2. Create a new JSP file as *fmt_formatDate*.



2. Rename the title as *Using JSTL tag library* and `<h2>` as *Use fmt_formatDate features*.

```
<title>fmt:parseDate feature</title>
</head>
<body>
  <h2>fmt:parseDate feature</h2>
  <!-- a Date time string -->
```

4. Before using JSTL's tag, we need to declare JSTL's library tag using Taglib directive.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
```

5. Add JSTL Taglib directive for JSTL core and JSTL formatting into the current code.
6. Add the following code in your page.

```
<c:set var="now" value="<%=new java.util.Date()%>" />
<p>
  Time (fmt:formatDate type="time"):
  <strong>
    <fmt:formatDate type="time" value="${now}" />
  </strong>
</p>
<p>
  Date (fmt:formatDate type="date"):
  <strong>
    <fmt:formatDate type="date" value="${now}" />
  </strong>
</p>
<p>
  Date, Time (fmt:formatDate type="both"):
  <strong>
    <fmt:formatDate type="both" value="${now}" />
  </strong>
</p>
<p>
  Date, Time Short (fmt:formatDate type="both" dateStyle="short"):
  <strong>
    <fmt:formatDate type="both" dateStyle="short" timeStyle="short" value="${now}" />
  </strong>
</p>
```

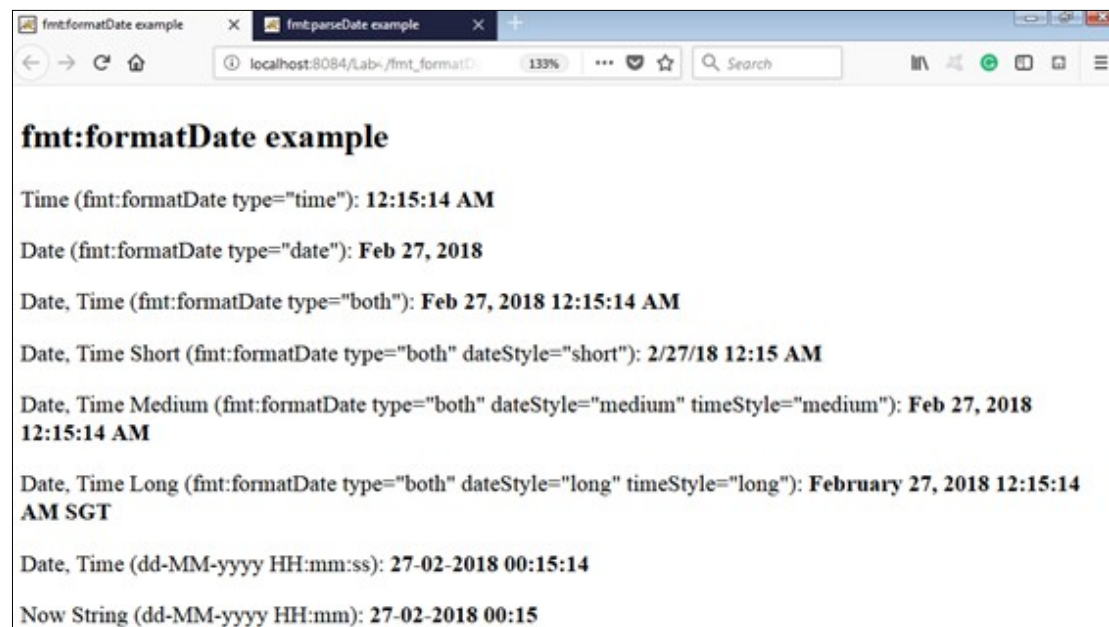
```

<p>
  Date, Time Medium (fmt:formatDate type="both" dateStyle="medium" timeStyle="medium"):
  <strong>
    <fmt:formatDate type="both" dateStyle="medium" timeStyle="medium" value="{now}" />
  </strong>
</p>
<p>
  Date, Time Long (fmt:formatDate type="both" dateStyle="long" timeStyle="long"):
  <strong>
    <fmt:formatDate type="both" dateStyle="long" timeStyle="long" value="{now}" />
  </strong>
</p>
<p>
  Date, Time (dd-MM-yyyy HH:mm:ss):
  <strong>
    <fmt:formatDate pattern="dd-MM-yyyy HH:mm:ss" value="{now}" />
  </strong>
</p>
<!-- Store in variable -->
<fmt:formatDate pattern="dd-MM-yyyy HH:mm" value="{now}" var="nowString"/>
<p>
  Now String (dd-MM-yyyy HH:mm):
  <strong>
    <c:out value="{nowString}" />
  </strong>
</p>

```

7. Compile and run *fmt_formatDate.jsp* page.

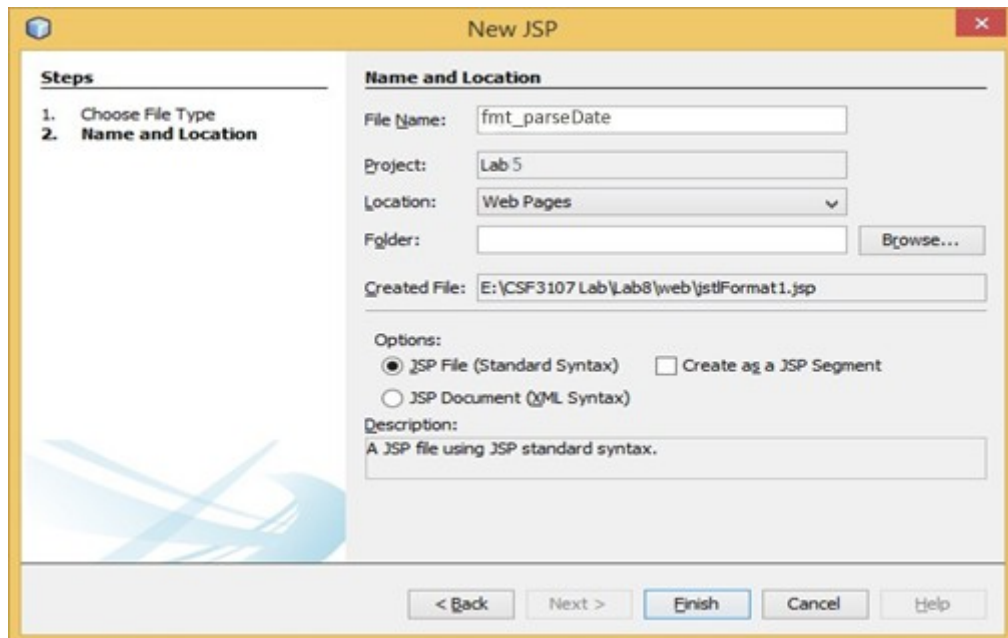
8. You will get the following output.



Task 2 - Using JSTL's fmt to Parse Date

Step 1

1. Create JSP's file and rename as *fmt_parseDate*.



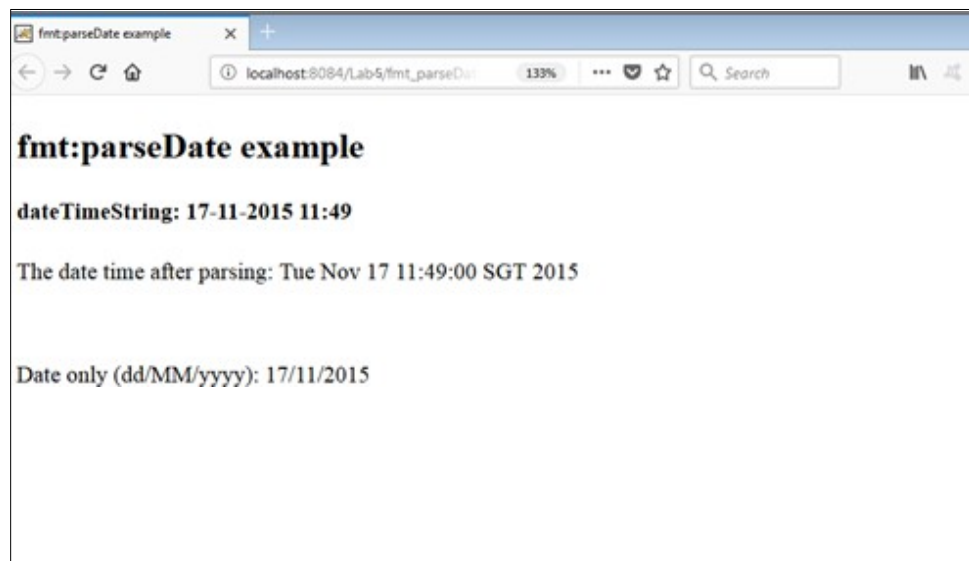
2. Add JSTL taglib directive for JSTL *core* and JSTL *formatting* into the current code.
3. Add the following code in your page.

```
<body>
  <c:set var="dateTimeString" value="17-11-2015 11:49" />
  <h4>
    dateTimeString:
    <c:out value="${dateTimeString}" />
  </h4>

  <!-- Parsing a date time string, and store in a variable type of java
  <fmt:parseDate value="${dateTimeString}"
    type="both" var="parsedDatetime" pattern="dd-MM-yyyy HH:mm" />
  <p>
    The date time after parsing:
    <c:out value="${parsedDatetime}" />
  </p>
  <br/>
  <p>
    Date only (dd/MM/yyyy):
    <fmt:formatDate value="${parsedDatetime}" pattern="dd/MM/yyyy" />
  </p>
</body>
```

5. Save *fmt_parseDate.jsp*
6. Compile and run *fmt_parseDate.jsp*.

7. You will get the following output.



Task 1

```
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
9 <%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14 <title>fmt:formatDate features</title>
15 </head>
16 <body>
17 <h2>fmt:formatDate features</h2>
18
19 <!-- Set the current date using JSTL -->
20 <c:set var="now" value="<%= new java.util.Date() %>" />
21
22 <p>
23 Time (fmt:formatDate type="time"):
24 <strong>
25 <fmt:formatDate type="time" value="{now}" />
26 </strong>
27 </p>
28 <p>
29 Date (fmt:formatDate type="date"):
30 <strong>
31 <fmt:formatDate type="date" value="{now}" />
32 </strong>
33 </p>
34 <p>
35 Date, Time (fmt:formatDate type="both"):
36 <strong>
37 <fmt:formatDate type="both" value="{now}" />
```

```
<fmt:formatDate type="both" value="{now}" />
</strong>
</p>
<p>
Date, Time Short (fmt:formatDate type="both" dateStyle="short" timeStyle="short"):
<strong>
<fmt:formatDate type="both" dateStyle="short" timeStyle="short" value="{now}" />
</strong>
</p>
<p>
Date, Time Medium (fmt:formatDate type="both" dateStyle="medium" timeStyle="medium"):
<strong>
<fmt:formatDate type="both" dateStyle="medium" timeStyle="medium" value="{now}" />
</strong>
</p>
<p>
Date, Time Long (fmt:formatDate type="both" dateStyle="long" timeStyle="long"):
<strong>
<fmt:formatDate type="both" dateStyle="long" timeStyle="long" value="{now}" />
</strong>
</p>
<p>
Date, Time (dd-MM-yyyy HH:mm:ss):
<strong>
<fmt:formatDate pattern="dd-MM-yyyy HH:mm:ss" value="{now}" />
</strong>
</p>
<fmt:formatDate pattern="dd-MM-yyyy HH:mm" value="{now}" var="nowString" />
<p>
Now String (dd-MM-yyyy HH:mm):
```

```

<fmt:formatDate pattern="dd-MM-yyyy HH:mm" value="
    <p>
      Now String (dd-MM-yyyy HH:mm):
      <strong>
        <c:out value="${nowString}" />
      </strong>
    </p>
  </body>
</html>

```

Task 2

```

6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
9 <%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
10 <!DOCTYPE html>
11 <html>
12 <head>
13   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14   <title>JSP Page</title>
15 </head>
16 <body>
17   <h2>fmt:parseDate example</h2>
18   <c:set var="dateTimeString" value="17-11-2015 11:49"/>
19   <h4>dateTimeString : <c:out value="${dateTimeString}"/></h4>
20
21   <!-- Parsing a date time string and storing in a variable of type Date -->
22   <fmt:parseDate value="${dateTimeString}" type="both" var="parsedDateTime" pattern="dd-MM-yyyy HH:mm" />
23   <p>The date time after parsing: <fmt:formatDate value="${parsedDateTime}" pattern="dd-MM-yyyy HH:mm" /></p>
24   <br/>
25   <p>Date only (dd/MM/yyyy): <fmt:formatDate value="${parsedDateTime}" pattern="dd/MM/yyyy" /></p>
26 </body>
27 </html>
28

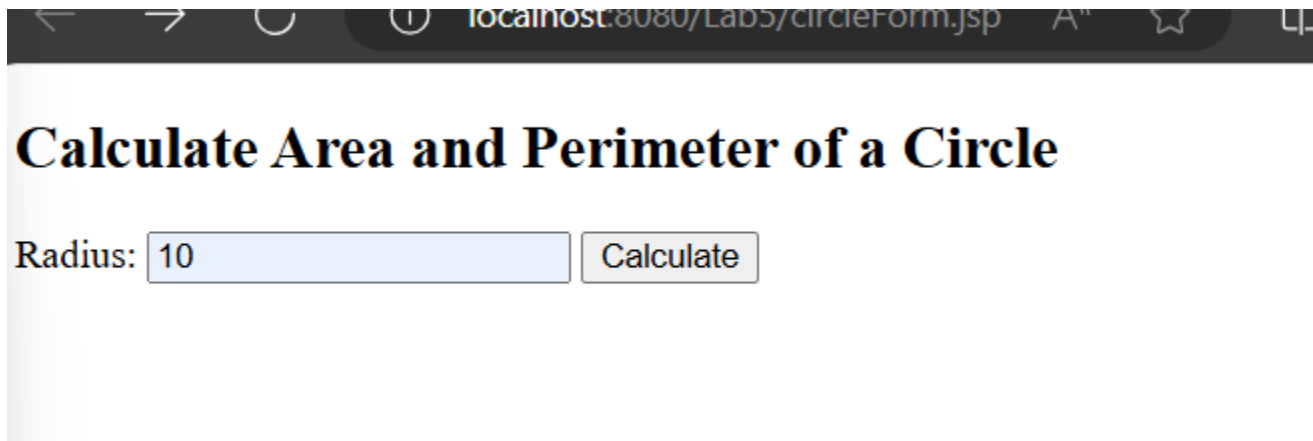
```

Reflection

1. What you have learnt from this exercise?

Exercise

1. Write a JSP's form that asks the user to key-in the radius of circle. The program should calculate the area of circle and the perimeter of circle. Finally, use JSTL library to format your result into 3 decimal places.



localhost:8080/Lab5/circleForm.jsp

Calculate Area and Perimeter of a Circle

Radius:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Circle Calculator</title>
  </head>
  <body>
    <h2>Calculate Area and Perimeter of a Circle</h2>
    <form action="calculateCircle.jsp" method="post">
      <label for="radius">Radius:</label>
      <input type="text" id="radius" name="radius" required />
      <input type="submit" value="Calculate" />
    </form>
  </body>
</html>
```

2. Rahim bought 800 shares of stock at a price of RM10.50 per share. He must pay her stock broker a 5 percent commission for the transaction. Write a web based program that calculates and displays the following:
- The amount paid for the stock alone without the commission.
 - The amount of the commission.
 - The total amount paid (for the stock plus the commission).

You should use JavaBeans to implement business logic and JSTL for display purposes.

stockTransaction.java

```
11 import java.io.Serializable;
12
13 public class StockTransaction implements Serializable {
14     private int shares;
15     private double pricePerShare;
16     private double commissionRate;
17
18     public StockTransaction() {
19         // Default constructor
20     }
21
22     // Getters and setters
23
24     public int getShares() {
25         return shares;
26     }
27
28     public void setShares(int shares) {
29         this.shares = shares;
30     }
31 }
```



```

31
32     public double getPricePerShare() {
33         return pricePerShare;
34     }
35
36     public void setPricePerShare(double pricePerShare) {
37         this.pricePerShare = pricePerShare;
38     }
39
40     public double getCommissionRate() {
41         return commissionRate;
42     }
43
44     public void setCommissionRate(double commissionRate) {
45         this.commissionRate = commissionRate;
46     }
47
48     // Methods to calculate transaction details
49
50     public double getStockCost() {
51         return shares * pricePerShare;
52     }
53
54     public double getCommission() {
55         return getStockCost() * commissionRate;
56     }
57
58     public double getTotalCost() {
59         return getStockCost() + getCommission();
60     }
61 }

```

calculateStock.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Stock Transaction Calculator</title>
</head>
<body>
    <h2>Stock Transaction Calculator</h2>
    <form action="transactionResult.jsp" method="post">
        <label for="shares">Number of Shares:</label>
        <input type="text" id="shares" name="shares" required><br>
        <label for="pricePerShare">Price per Share (RM):</label>
        <input type="text" id="pricePerShare" name="pricePerShare" required><br>
        <input type="submit" value="Calculate">
    </form>
</body>
</html>

```

transactionResult.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<jsp:useBean id="transaction" class="com.mycompany.StockTransaction" scope="request" />
<jsp:setProperty name="transaction" property="shares" value="${param.shares}" />
<jsp:setProperty name="transaction" property="pricePerShare" value="${param.pricePerShare}" />
<jsp:setProperty name="transaction" property="commissionRate" value="0.05" />

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Stock Transaction Results</title>
</head>
<body>
    <h2>Stock Transaction Results</h2>
    <p>Amount paid for the stock alone: RM <fmt:formatNumber value="${transaction.stockCost}" type="currency" /></p>
    <p>Amount of the commission: RM <fmt:formatNumber value="${transaction.commission}" type="currency" /></p>
    <p>Total amount paid (for the stock plus the commission): RM <fmt:formatNumber value="${transaction.totalCost}" type="currency" /></p>
</body>
</html>

```