CSM3023 WEB BASED APPLICATION DEVELOPMENT (K1)

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS

SEMESTER II 2023/2024

LAB 5 – JSP: JavaBeans & Java Standard Tag Library (JSTL)

Prepared for:

DR. MOHAMAD NOR HASSAN

Prepared by:

MUHAMMAD IZZUL WAFIY BIN IZAM (S65466)

Task 1

Message1.jsp

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page language="java"%>
<%@page info="Using JSP Standard Action to call JavaBeans"%>
<%@page import="java.util.Date, lab5.com.Message"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Using JSP Scriplet</title>
    </head>
    <body>
        <h1>Using JSP Scriplet to call JavaBeans</h1>
        <%
            //Create an object..
            Message objMsg = new Message();

            //Assign value..
            objMsg.setMsg("Welcome to CSM3023 course...!");

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

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

Message.java

```java
package lab5.com;


public class Message {
    private String msg;

    public String getMsg() {
        return msg;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }
}
```

Output:

Using JSP Scriplet to call JavaBeans

Welcome to CSM3023 course...!

Current date is Tue May 14 23:19:49 MYT 2024

Reflection

1. What you have learnt from this exercise?
I learned how to create and use a JavaBean in a JSP page, utilize JSP scriptlets to manipulate JavaBean objects, and display dynamic content such as messages and dates in a JSP.

2. Explain the differences when calling JavaBeans using JSP Standard Action and Java Scriptlet.
JSP Standard Actions use XML-like tags that keep Java code out of the JSP, making it cleaner and easier to maintain, while JSP Scriptlets embed Java code directly within the JSP, providing more flexibility but potentially making the code harder to read and maintain.

Task 2

registerTraining.jsp

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="lab5.com.TrainingRegistration"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Register for Training</title>
    <script>
        function validateForm() {
            var icNo = document.forms["registrationForm"]["icNo"].value;
            var name = document.forms["registrationForm"]["name"].value;
            var trainingType = document.forms["registrationForm"]["trainingType"].value;
            var numOfPax = document.forms["registrationForm"]["numOfPax"].value;
            var isStudent = document.forms["registrationForm"]["isStudent"].value;

            // Check if any field is empty
            if (icNo === "" || name === "" || trainingType === "" || numOfPax === "" || isStudent === "") {
                alert("Please fill in all fields");
                return false;
            }

            // Check if number of pax is a valid number
            if (isNaN(numOfPax)) {
                alert("Number of pax must be a valid number");
                return false;
            }

            // Check if number of pax is not negative
            if (numOfPax < 0) {
                alert("Number of pax cannot be negative");
                return false;
            }

            return true;
        }
    </script>
</head>
<body>
    <h1>Register for Training</h1>
    <form name="registrationForm" action="processTraining.jsp" onsubmit="return validateForm()" method="post">
        <label for="icNo">IC No:</label>
        <input type="text" id="icNo" name="icNo"><br><br>

        <label for="name">Name:</label>
        <input type="text" id="name" name="name"><br><br>

        <label for="trainingType">Type of Training:</label>
        <select id="trainingType" name="trainingType">
            <option value="">Select Training</option>
            <option value="1">C++ training</option>
            <option value="2">Java for beginner</option>
            <option value="3">HTML5</option>
            <option value="4">Java EEE</option>
            <option value="5">Android Programming</option>
        </select><br><br>

        <label for="numOfPax">Number of Pax:</label>
        <input type="text" id="numOfPax" name="numOfPax"><br><br>


        <label>Student:</label>
        <input type="radio" id="isStudentYes" name="isStudent" value="1">
        <label for="isStudentYes">Yes</label>
        <input type="radio" id="isStudentNo" name="isStudent" value="0">
        <label for="isStudentNo">No</label><br><br>

        <%!
    public String getTrainingTypeName(String trainingType) {
        switch (trainingType) {
            case "1":
                return "C++ training";
            case "2":
                return "Java for beginner";
            case "3":
                return "HTML5";
            case "4":
                return "Java EEE";
            case "5":
                return "Android Programming";
        }
    }
%>

        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

TrainingRegistration.java

```java
package lab5.com;

public class TrainingRegistration {
    private String icNo;
    private String name;
    private String trainingType;
    private int numOfPax;
    private boolean isStudent;

    // Getters and setters
    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 getTrainingType() {
        return trainingType;
    }

    public void setTrainingType(String trainingType) {
        this.trainingType = trainingType;
```

```java
}

public int getNumOfPax() {
    return numOfPax;
}

public void setNumOfPax(int numOfPax) {
    this.numOfPax = numOfPax;
}

public boolean isStudent() {
    return isStudent;
}

public void setStudent(boolean isStudent) {
    this.isStudent = isStudent;
}

// Method to calculate total fee
public double calculateTotalFee() {
    double fee = 0;

    switch (trainingType) {
        case "1": // C++ training
            fee = 3000 * numOfPax;
            break;
        case "2": // Java for beginner
            fee = 3000 * numOfPax;
            break;
        case "3": // HTML5
            fee = 2800 * numOfPax;
            break;
```

TrainingRegistration.java

```
    public double calculateTotalFee() {
        double fee = 0;

        switch (trainingType) {
            case "1": // C++ training
                fee = 3000 * numOfPax;
                break;
            case "2": // Java for beginner
                fee = 3000 * numOfPax;
                break;
            case "3": // HTML5
                fee = 2800 * numOfPax;
                break;
            case "4": // Java EE
                fee = 5500 * numOfPax;
                break;
            case "5": // Android Programming
                fee = 3200 * numOfPax;
                break;
            default:
                break;
        }

        // Apply discount for students
        if (isStudent) {
            fee *= 0.9; // 10% discount
        }

        return fee;
    }
}
```

processTraining.jsp

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="lab5.com.TrainingRegistration"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Processing Training Registration</title>
</head>
<body>
<%
    // Retrieve form data
    String icNo = request.getParameter("icNo");
    String name = request.getParameter("name");
    String trainingType = request.getParameter("trainingType");
    int numOfPax = Integer.parseInt(request.getParameter("numOfPax"));
    boolean isStudent = request.getParameter("isStudent").equals("1");

    // Create an instance of TrainingRegistration
    TrainingRegistration registration = new TrainingRegistration();
    registration.setIcNo(icNo);
    registration.setName(name);
    registration.setTrainingType(trainingType);
    registration.setNumOfPax(numOfPax);
    registration.setStudent(isStudent);

    // Calculate total fee
    double totalFee = registration.calculateTotalFee();
%>

<h1>Training Registration Acknowledgement</h1>
<p>IC No: <%= registration.getIcNo() %></p>
<p>Name: <%= registration.getName() %></p>
<p>Type of Training: <%= registration.getTrainingType()%></p>
<p>Number of Pax: <%= registration.getNumOfPax() %> <%= "person/s" %></p>
<p>Student: <%= registration.isStudent() ? "Yes" : "No" %></p>
<p>Total Fee: RM <%= totalFee %></p>

</body>
</html>
```

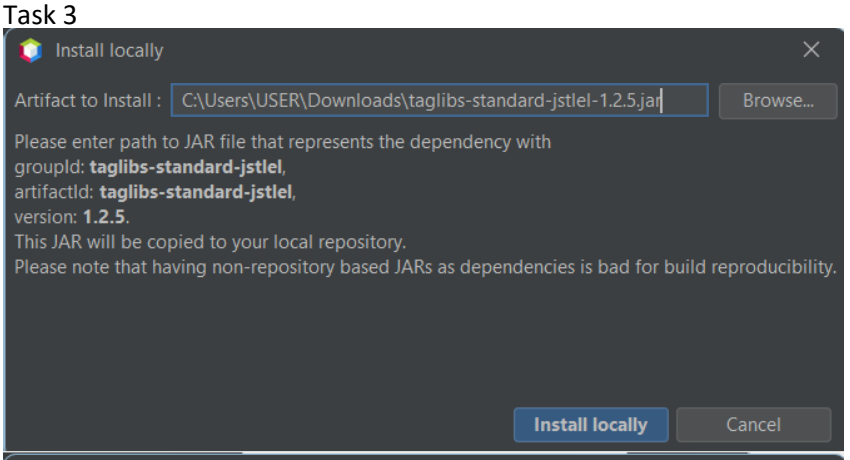
Output:

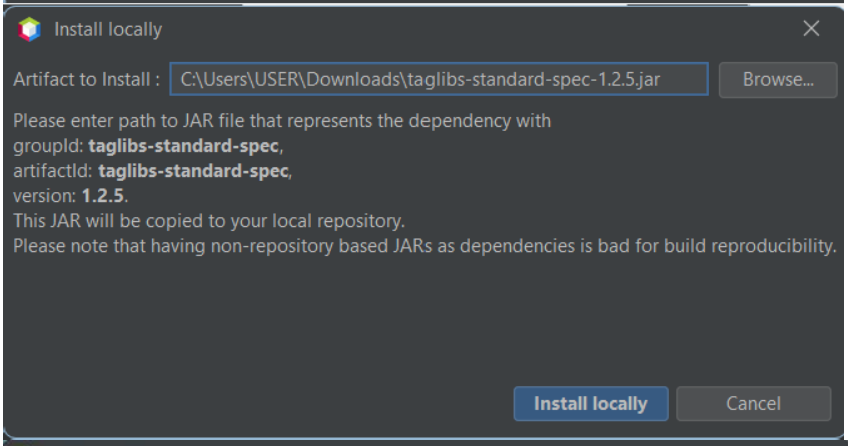
Reflection

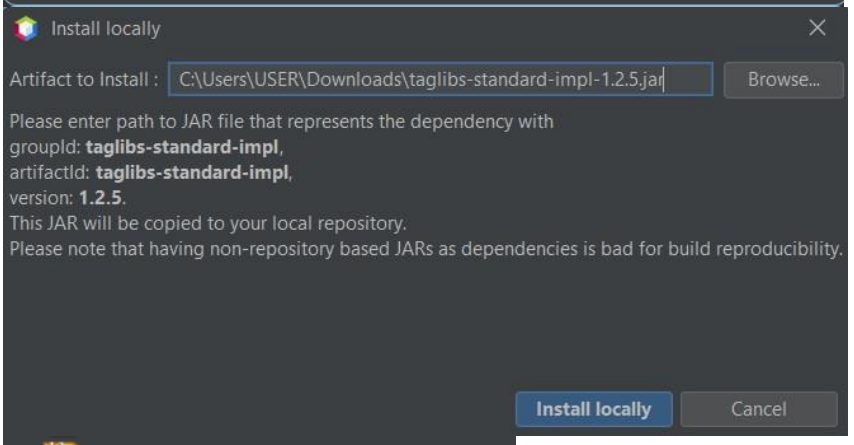
1. What you have learnt from this exercise?
I learned how to create and use a JavaBean in a JSP page, validate user input with JavaScript, and process form data to display dynamic content.

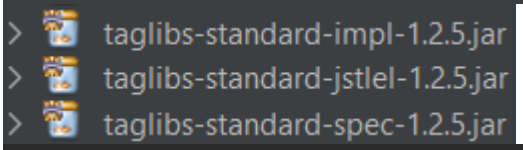
2. Describe the steps how you construct Register JavaBeans?
Define a package, create a public JavaBean class, declare private fields, generate public getter and setter methods, optionally add business logic methods, compile the JavaBean, and use it in a JSP page.

Task 3










```
<dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>2.0.0</version>
    <scope>compile</scope>
</dependency>
```

Reflection

What you have learnt from this exercise?
I had learned how to add JSTL in dependencies.

Task 4
1.
jstlCore1.jsp

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri= "http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<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>
        <c:set var="message" value="Welcome to CSM3023 - Web Programming course..!" />
        <p> <c:out value="${message}"/></p>
    </body>
</html>
```

Output:

Use JSTL's features

Welcome to CSM3023 - Web Programming course..!

2.
userRegistration.html

```
<!DOCTYPE html>
...4 lines
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Registration</title>
</head>
<body>
    <form action="processUser.jsp" method="POST">
        <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" required>
        <label for="male">Male</label>
        <input type="radio" id="female" name="gender" value="female" required>
        <label for="female">Female</label><br><br>

        <label for="userType">Type of User:</label>
        <select id="userType" name="userType">
            <option value="beginner">Beginner</option>
            <option value="intermediate">Intermediate</option>
            <option value="advanced">Advanced</option>
        </select><br><br>

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

        <input type="submit" value="Submit">
        <input type="button" value="Cancel">
    </form>
    <footer>
        &copy Haris Zakuwan
    </footer>
</body>
</html>
```

processUser.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Information</title>
</head>
<body>
    <h2>Retrieve info using request parameters & display it using JSP expression</h2>

    <%
        String name = request.getParameter("name");
        String surname = request.getParameter("surname");
        String password = request.getParameter("password");
        String gender = request.getParameter("gender");
        String userType = request.getParameter("userType");
        String preferLanguage = request.getParameter("preferLanguage");
    %>

    <p><b>First Name:</b> <%= name %></p>
    <p><b>Surname:</b> <%= surname %></p>
    <p><b>Password:</b> <%= password %></p>
    <p><b>Gender:</b> <%= gender %></p>
    <p><b>Type of user:</b> <%= userType %></p>
    <p><b>Prefer Language:</b> <%= preferLanguage %></p>
</body>
</html>
```

Output:

Name: Rishhh

Surname: Zuzu

Password: •••

Gender: ◉ Male ○ Female

Type of User: Advanced

Prefer Language: ☑ Malay ☐ English ☐ Mandarin ☐ Tamil

Submit Cancel

© Haris Zakuwan

Retrieve info using request parameters & display it using JSP expression

First Name: Rishhh

Surname: Zuzu

Password: 123

Gender: male

Type of user: advanced

Prefer Language: malay

3.
jstlFormat1.jsp

```jsp
<%@ page contentType="text/html; charset=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>JSP Page</title>
    </head>
    <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>
        <p>Formatting number to 3 decimal places : <fmt:formatNumber type="number" maxFractionDigits="3" value="${total}"/></p>
        <p>Formatting as percentage : <fmt:formatNumber type="percent" value="${total}"/></p>
    </body>
</html>
```

Output:

Using JSTL formatting tag for formatting

Number to be formatted is 2880.4638

Formatting number as currency with currency code : MYR2,880.46

Formatting number to the nearest 2 integer digit : 80.464

Formatting number by grouping : 2,880.464

Formatting number to 3 decimal places : 2,880.464

Formatting as percentage : 288,046%

Reflection

1. What the purpose of using JSTL's tag library?
The purpose of using JSTL is to simplify the development of JSP pages by providing a set of standard tags for common tasks, such as iteration, conditionals, XML processing, internationalization, and database access, thereby reducing the amount of Java code embedded in JSPs and enhancing code readability and maintainability.

2. List FIVE(5) categories of JSTL library.
- Core tags (c)
- Formatting Tags (fmt)
- SQL Tags (sql)
- XML Tags (x)
- Function Tags (fn)

Task 5

1.

fmt_formatDate.jsp

```jsp
<%@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>fmt:parseDate feature</title>
    </head>
    <body>
        <h2>fmt:parseDate feature</h2>

        <!-- Declare JSTL's taglib directives -->
        <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
        <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>

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

Output:

fmt:parseDate feature

Time (fmt:formatDate type="time"): **12:20:44 AM**

Date (fmt:formatDate type="date"): **May 15, 2024**

Date, Time (fmt:formatDate type="both"): **May 15, 2024, 12:20:44 AM**

Date, Time Short (fmt:formatDate type="both" dateStyle="short"): **5/15/24, 12:20 AM**

Date, Time Medium (fmt:formatDate type="both" dateStyle="medium" timeStyle="medium"): **May 15, 2024, 12:20:44 AM**

Date, Time Long (fmt:formatDate type="both" dateStyle="long" timeStyle="long"): **May 15, 2024, 12:20:44 AM MYT**

Date, Time (dd-MM-yyyy HH:mm:ss): **15-05-2024 00:20:44**

Now String (dd-MM-yyyy HH:mm): **15-05-2024 00:20**

2.
fmt_parseDate

```jsp
<%@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>fmt:parseDate example</title>
    </head>
    <body>
        <h2>fmt:parseDate example</h2>
        <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>
</html>
```

Output:

fmt:parseDate example

dateTimeString: 17-11-2015 11:49

The date time after parsing: Tue Nov 17 11:49:00 MYT 2015

Date only (dd/MM/yyyy) : 17/11/2015

Reflection

1. What you have learnt from this exercise?

I learned how to use JSTL tags to format and parse dates and times in JSP pages.

Exercise

1.

circle.jsp

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

Calculate.jsp

```jsp
<%@ page contentType="text/html; charset=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>Calculate Circle Area and Perimeter</title>
</head>
<body>
    <h2>Results</h2>
    <%-- Retrieve radius from request --%>
    <c:set var="radius" value="${param.radius}" />
    <%-- Calculate area and perimeter --%>
    <c:set var="area" value="${3.14159265 * radius * radius}" />
    <c:set var="perimeter" value="${2 * 3.14159265 * radius}" />

    <p>Area of the circle: <fmt:formatNumber value="${area}" pattern="###.###"/></p>
    <p>Perimeter of the circle: <fmt:formatNumber value="${perimeter}" pattern="###.###"/></p>
</body>
</html>
```

Output:

Calculate Circle Area and Perimeter

Enter the radius of the circle: 20 Calculate

Results

Area of the circle: 1256.637

Perimeter of the circle: 125.664

2.
Brokerage.jsp

```jsp
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="lab5.com.processBrokerage"%>
<%@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>LocoTex Trading Broker</title>
    </head>
    <body>
        <h1>Welcome to LocoTex Trading Broker!</h1>
        <%
            //initialize all variables needed
            int shares = 800;
            double price = 10.50;
            //instantiate broker object
            processBrokerage broker = new processBrokerage(shares, price);
        %>

        <!-- declare variable using taglibs approach (c:set)-->
        <c:set var="amount" value="<%=broker.getAmountB(shares, price)%>" />
        <c:set var="commission" value="<%=broker.getCommission(shares, price)%>" />
        <c:set var="total" value="<%=broker.getAmountA(shares, price)%>" />

        <!-- output the value using fmt:format to ensure the output is 2 decimal places -->
        <p>Amount (without commission): RM <fmt:formatNumber type="number" minFractionDigits="2" value="${amount}" /></p>
        <p>Commission charged: RM <fmt:formatNumber type="number" minFractionDigits="2" value="${commission}" /></p>
        <p>Total amount paid (commission included): RM <fmt:formatNumber type="number" minFractionDigits="2" value="${total}" /></p>
    </body>
</html>
```

processBrokerage.java

```java
package lab5.com;

/**
 *
 * @author USER
 */
public class processBrokerage {
    private int shares;
    private double price;

    public processBrokerage() {
    }

    public processBrokerage(int shares, double price) {
        this.shares = shares;
        this.price = price;
    }

    public int getShares() {
        return shares;
    }

    public void setShares(int shares) {
        this.shares = shares;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public double getAmountB(int shares, double price){
        double amountB = shares * price;
        return amountB;
    }
    public double getCommission(int shares, double price){
        double commission;
        commission = shares * price * 0.05;
        return commission;
    }

    public double getAmountA(int shares, double price){
        double amountA = shares * price;
        double commission = shares * price * 0.05;
        return amountA + commission;
    }
}
```

Output:

Welcome to LocoTex Trading Broker!

Amount (without commission): RM 8,400.00

Commission charged: RM 420.00

Total amount paid (commission included): RM 8,820.00