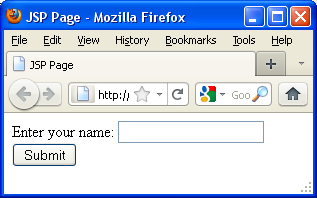
44-563 Developing Web Applications and Services Spring 2013

Exam 3

Part 1 JavaServer **PAGES**

1. (10 pts) Write the code for the body of the welcome page shown below. When the button is clicked, the response page "GreetUser.jsp" is returned. Write only the code that goes inside the <body> tag of the welcome page.



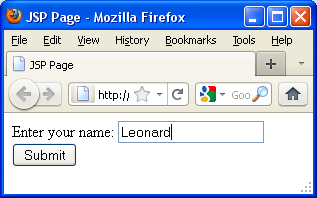
**<form action="GreetUser.jsp">**

**Enter your name: <input type="text" name="userName" /><br />**

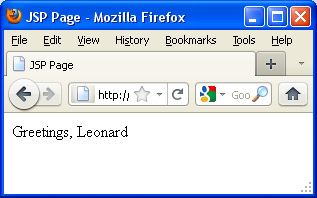
**<input type="submit" value="Submit" />**

**</form>**

2. (7 pts) (Continuation of Problem 1.) The response page displays "Greetings, " followed by the user name that was entered in the welcome page. For example, if the user types "Leonard" in the input box,



the response page looks like this:



Write the code for the body of the response page. You MUST use EL to obtain the user name. Write only the code that goes inside the <body> tag of the response page.

**Greetings, ${param.userName}**

3. (9 pts) Assume class **Movie** has the constructor shown below:

**public Movie(String title, int rating)**

In an MVC application, the controller

* creates a **Movie** object with title "The Matrix" and rating 4,
* makes the object the value of a request attribute named **scifiMovie**
* forwards to the view

Complete the code necessary to do this. **Note: You must write three lines of code.**

**// Create Movie object**

**Movie matrix = new Movie("The Matrix", 4);**

**// Set the request attribute**

**request.setAttribute("scifiMovie", matrix);**

**RequestDispatcher view**

**= request.getRequestDispatcher("/moviePage.jsp");**

**// Forward to the view**

**view.forward(request, response);**

4. (6 pts) (Continuation of Problem 3.) Write the code that would be included in the response page moviePage.jsp to display the following:

Title: The Matrix

Rating: 4

Write only the code necessary to display the text shown above. **You must use EL expressions to obtain the title and the rating of the movie**.

**Title: ${scifiMovie.title}<br />**

**Rating: ${scifiMovie.rating}**

5. (8 pts) Assume the value of param.title is the title of a movie. Write code to be included in a JSP to display

* Dream if the title is Inception
* Lizard if the title is Rango
* Boring for all other titles

Use the JSTL <c:choose> tag to write your code.

**<c:choose>**

**<c:when test="${param.title eq 'Inception'}">**

**Dream**

**</c:when>**

**<c:when test="${param.title eq 'Rango'}">**

**Lizard**

**</c:when>**

**<c:otherwise>**

**Boring**

**</c:otherwise>**

**</c:choose>**

6. (10 pts) Assume bookList is an array of Strings. Each entry in the array is the title of a book. A servlet executes the statement below and then forwards the request to a JavaServer Page.

**request.setAttribute("books",** **bookList);**

Complete the code below that would be included in a JavaServer Page to display the list of book titles in an **HTML table**. Use a JSTL <c:forEach> tag.

**<h2>Book List:</h2><br>**

**<table>**

**<c:forEach var="title" items="${books}">**

**<tr>**

**<td>${title}</td>**

**</tr>**

**</c:forEach>**

**</table>**

Part 2 JavaServer **FACES**

7. (4 pts) The welcome page index.xhtml for a JSF application contains the following code for a button inside a form:

**<h:commandButton value="Click Me" action="CheckOut" />**

Complete the code so that the application navigates to the page CheckOut.xhtml when the button is clicked.

8. (4 pts) The code below appears inside a form element on a JSF page.

<h:inputText value="#{UserBean.studentGpa}" />

(a) What method is invoked by the expression #{UserBean.studentGpa} when the web page is displayed?

A. studentGpa()

B. getStudentGpa()

C. setStudentGpa()

(b) What method is invoked by the expression #{UserBean.studentGpa} when the form is submitted?

A. studentGpa()

B. getStudentGpa()

C. setStudentGpa()

9. (6 pts) Suppose we want to internationalize a web application so that it can display text in either English or French. We have declared a resource bundle in faces-config.xml:

**<application>**

**<resource-bundle>**

**<base-name>**

MyText

**</base-name>**

**<var>**txt**</var>**

**</resource-bundle>**

**</application>**

**<application>**

**<locale-config>**

**<default-locale>**

en\_US

**</default-locale>**

**<supported-locale>**

fr\_FR

**</supported-locale>**

**</locale-config>**

**</application>**

(a) We need a label **title** with text "Movie title" (or "Titre du film" in French).

Write the entry that must be included in the file MyText.properties.

**title=Movie title**

Write the entry that must be included in the file MyText\_fr\_FR.properties.

**title = Titre du film**

(b) Complete the JSF code below that displays the label.

**<h:outputText value="#{txt.title}"**

10. (4 pts) The code below is used to input a movie's rating in a text box.

**<h:inputText**

**value="#{movieBean.movieRating}"**

**id="ratingField"**

**/>**

Rewrite the code so that the user is required to enter a value for this field.

**<h:inputText**

**value="#{movieBean.movieRating}"**

**id="ratingField"**

**required="true"**

**/>**

11. (10 pts) Rewrite the original code in Problem 10 to validate that the user enters a value between 1 and 5 (inclusive).

**<h:inputText**

**value="#{movieBean.movieRating}"**

**id="ratingField">**

**<f:validateLongRange**

**minimum="1"**

**maximum="5" />**

**</h:inputText>**

12. (6 pts) Rewrite the code below by adding a converter so that priceOfTicket will be displayed as currency.

**<h:outputText value="#{movieBean.priceOfTicket}">**

**<f:convertNumber type="currency" />**

**</h:outputText>**

13. (8 pts) One of the examples in the JSF unit displayed a grade report for students in a class. It used a dataTable element to display the report. The beginning of the code for the dataTable is shown below. Write the code to create a column in the dataTable that displays the names of the students. The column heading must be “Name”.

**<h:dataTable value=”#{gradeBook.students}” var=”student”>**

**<h:column>**

**<f:facet name=”header”>Name</f:facet>**

**#{student.name}**

**</h:column>**

14. (8 pts) Here is some code from a JSF template:

**<h:body>**

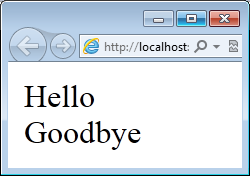
**<ui:insert name="greeting"></ui:insert>**

**<br />**

**<ui:insert name="salutation"></ui:insert>**

**</h:body>**

Complete the code in the facelet below so that the web page produced looks like the following when displayed in a browser.



**<h:body>**

**<ui:composition template="./template.xhtml">**

**<ui:define name="greeting">**

**Hello**

**</ui:define>**

**<ui:define name="salutation">**

**Goodbye**

**</ui:define>**

**</ui:composition>**

**</h:body>**