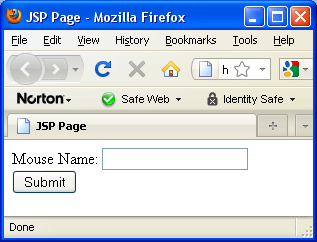
44-563 Web Services Technology Fall 2010

Exam 3

Part 1 JavaServer **PAGES**

1. (12 pts) Write the code for the body of the welcome page shown below. When the button is clicked, the response page "Hello.jsp" is returned. Write only the code that goes inside the <body> tag of the welcome page.



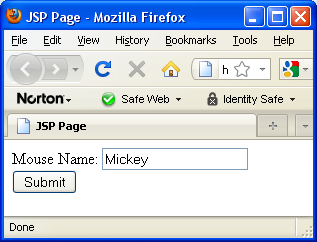
**<form action="Hello.jsp">**

**Mouse Name: <input type="text" name="mouseName" /><br />**

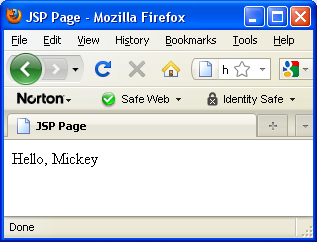
**<input type="submit" value="Submit" />**

**</form>**

2. (10 pts) (Continuation of Problem 1.) The response page displays "Hello, " followed by the mouse name that was entered in the welcome page. For example, if the user types "Mickey" 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 mouse name. Write only the code that goes inside the <body> tag of the response page.

**Hello, ${param.mouseName}**

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

**public Student(String name, int id)**

In an MVC application, the controller

* creates a **Student** object with name "Carmen" and id 919191919,
* makes the object the value of a request attribute named **honorStudent**
* forwards to the view

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

**// Create Student object**

**Student carmen = new Student("Carmen", 919191919);**

**// Set the request attribute**

**request.setAttribute("honorStudent", carmen);**

**RequestDispatcher view**

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

**// Forward to the view**

**view.forward(request, response);**4. (6 pts) The class Canine is defined as shown below:

**public class Canine**

**{**

**private String name;**

**private String major;**

**public Canine(String name, String major)**

**{**

**this.name = name;**

**this.major = major;**

**}**

**public String getName()**

**{**

**return name;**

**}**

**public String getMajor()**

**{**

**return major;**

**}**

**}**

A servlet executes the following code:

**Canine smartDog = new Canine("Leonard", "Math");**

**request.setAttribute("poodle", smartDog);**

and then forwards to a JSP. **Write an EL expression** that could be included in the JSP to display the canine's major.

**${poodle.major}**

5. (12 pts) Assume the value of param.creature is the name of an animal. Write code to be included in a JSP to display

* Dog if the name is Leonard
* Cat if the name is Maude
* Mouse for all other names

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

**<c:choose>**

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

**Dog**

**</c:when>**

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

**Cat**

**</c:when>**

**<c:otherwise>**

**Mouse**

**</c:otherwise>**

**</c:choose>**

6. (12 pts) Assume catList is an array of Strings. Each entry in the array is the name of a cat. A servlet executes the statement below and then forwards the request to a JavaServer Page.

**request.setAttribute("cats",** **catList);**

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

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

**<table>**

**<c:forEach var="cat" items="${cats}">**

**<tr>**

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

**</tr>**

**</c:forEach>**

**</table>**

Part 2 JavaServer **FACES**

7. (5 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="Hi" />

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

8. (5 pts) In the code below

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

the expression #{UserBean.studentGpa} invokes what method?

A. studentGpa()

B. getStudentGpa()

C. setStudentGpa()

9. (7 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>**labels**</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 **student** with text "Student name" (or " Nom de l'étudiant" in French).

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

**student=Student name**

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

**student=Nom de l'étudiant**

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

**<h:outputText value="#{labels.student}" />**

10. (5 pts) The code below is used to input a dog's age in a text box.

**<h:inputText**

**value="#{UserBean.dogAge}"**

**id="ageField"**

**/>**

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

**<h:inputText**

**value="#{UserBean.dogAge}"**

**id="ageField"**

**required="true"**

**/>**

11. (10 pts) Rewrite the original code in Problem 10 to validate that the user enters a value between 2 and 18 (inclusive).

**<h:inputText value="#{UserBean.dogAge}"**

**id="ageField">**

**<f:validateLongRange**

**minimum="2"**

**maximum="18" />**

**</h:inputText>**

12. (7 pts) Rewrite the code below by adding a converter so that totalCost will be displayed as currency.

<h:outputText value="#{carBean.totalCost}">

<f:convertNumber type="currency" />

</h:outputText>

**ADDITIONAL SPACE**