44-563 Web Services Technology Exam 2 Spring 2010

1. (8 pts) Complete the code in the processRequest() method of a servlet that returns a web page that displays the message Hello, Web Surfer. The title of the page must be Problem One.

**PrintWriter out = response.getWriter();**

**try {**

**out.println("<html>");**

**out.println("<head>");**

**out.println("<title>Problem One</title>");**

**out.println("</head>");**

**out.println("<body>");**

**out.println("<h1>Hello, Web Surfer</h1>");**

**out.println("</body>");**

**out.println("</html>");**

**} finally {**

**out.close();**

**}**

*In problems 2 and 3, you will write XML code to be included in the deployment descriptor for a web application. Here are some of the tags you might need to use:*

**<context-param> <description>**

**<init-param> <listener>**

**<listener-class> <param-name>**

**<param-value> <servlet>**

**<servlet-class> <servlet-name>**

**<servlet-mapping> <url-pattern>**

2. (5 pts) Write the XML code that would be included in the deployment descriptor to define a context init parameter named **dogName** with value **Eve**.

**<context-param>**

**<param-name>dogName</param-name>**

**<param-value>Eve</param-value>**

**</context-param>**

3. (10 pts) Write the XML code that would be included in the deployment descriptor for a servlet with name **Cat Servlet**, class **Kitty**, and url **/Cat**. The servlet has an init parameter named **catName** with value **Moskowitz**.

**<servlet>**

**<servlet-name>Cat Servlet</servlet-name>**

**<servlet-class>Kitty</servlet-class>**

**<init-param>**

**<param-name>catName</param-name>**

**<param-value>Moskowitz</param-value>**

**</init-param>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>Cat Servlet</servlet-name>**

**<url-pattern>/Cat</url-pattern>**

**</servlet-mapping>**

4. (a) (4 pts) Write servlet code that would assign the value of the request parameter named **sn** to the String variable **snakeName**.

**String snakeName = request.getParameter("sn");**

(b) (4 pts) Write servlet code that would assign the value of the servlet init parameter named **fn** to the String variable **fishName**.

**String fishName**

**= getServletConfig().getInitParameter("fn");**

(c) (4 pts) Write servlet code that would assign the value of the context init parameter named **bn** to the String variable **birdName**.

**String birdName**

**= getServletContext().getInitParameter("bn");**

5. (5 pts) Assume a client invokes servlet S1 that sets up a session attribute as follows:

**HttpSession session = request.getSession();**

**session.setAttribute("city", "New York");**

Later the same client invokes another servlet S2. Write the code that must be included in servlet S2 to assign the value of the session attribute to the variable **cityName**.

**HttpSession session = request.getSession();**

**String cityName**

**= (String)session.getAttribute("city");**

6. (9 pts) Assume class Dog has the constructor shown below:

**public Dog(String dogName, int dogAge)**

In an MVC application, the controller

* creates a Dog object with name "Carmen" and age 7,
* makes the object the value of a request attribute named **dog**
* forwards to the view

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

**// Create Dog object**

**Dog d = new Dog("Carmen", 7);**

**// Set the request attribute**

**request.setAttribute("dog", d);**

**RequestDispatcher view**

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

**// Forward to the view**

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

7. (5 pts) Assume we have a request parameter named "dogName". Write an EL expression that could be included in a JSP to obtain the dog's name.

**${param.dogName}**

8. (6 pts) The class Car is defined as shown below:

**public class Car**

**{**

**private String make;**

**private String model;**

**public Car(String make, String model)**

**{**

**this.make = make;**

**this.model = model;**

**}**

**public String getMake()**

**{**

**return make;**

**}**

**public String getModel()**

**{**

**return model;**

**}**

**}**

A servlet executes the following code:

**Car theCar = new Car("Toyota", "Prius");**

**request.setAttribute("car", theCar);**

**RequestDispatcher view**

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

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

Write code that could be included in a JSP to display the car's make and model. **You must use EL to do this**.

**The make of the car is ${car.make}**

**The model of the car is ${car.model}**

9. (5 pts) Assume a welcome page contains this form:

**<form action="SayHi.jsp"** **method="post">**

**Movie: <input type="text"** **name="movieName"/><br>**

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

**</form>**

If the user enters **Armagedon** for the movie, the response must be

**Armagedon**

**Hello, asteroid**

Otherwise, the response is only the name of the movie.

Write code that would be included in **SayHi.jsp** to produce this response. You must use the JSTL **<c:if>** tag and EL. Do NOT write the complete .jsp file—just show the code necessary to produce the response described above.

**${param.movieName}<br>**

**<c:if test="${param.movieName eq 'Armagedon'}">**

**Hello, asteroid**

**</c:if>**

10. (5 pts) Assume the value of **param.movie** is the name of a movie. Write code to be included in a .jsp file to display

* **Neytiri** if the movie is **Avatar**,
* **Neo** if the movie is **The Matrix**, and
* **Arnold** for all other movies.

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

**<c:choose>**

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

**Neytiri**

**</c:when>**

**<c:when test="${param.movie eq 'The Matrix'}">**

**Neo**

**</c:when>**

**<c:otherwise>**

**Arnold**

**</c:otherwise>**

**</c:choose>**11. (5 pts) Assume **movieList** is an array of Strings. Each entry in the array is a movie title. A servlet executes the statement below and then forwards the request to a Java Server Page.

**request.setAttribute("movies",** **movieList);**

Complete the code below that would be included in the Java Server Page to display the list of movie titles in an unordered list. Use a JSTL **<c:forEach>** tag.

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

**<ul>**

**<c:forEach var="movie" items="${movies}">**

**<li>${movie}<li>**

**</c:forEach>**

**</ul>**