**JSP Assignment Questions: Part 1**

1. Write JSP Script which accepts user name and nick name from user. At first visit, display message “Hello user name” and for next successive requests, display “Hello nick name”. Use username if visit count is odd and nick name if visit count is even. (use declaration scripting elements)
2. Write JSP code to generate department wise monthly attendance reports of employee.
3. Write JSP code to display list of blood donors with details for particular blood group.
4. Write JSP code to search a student placement details from the placement database by student id.
5. Write JSP code to accept product details and store in database table. If the product is already inserted, display the appropriate message to insert another product.
6. Write JSP code which adds the student education details in a database through Java bean. Also display the student details who have secured first class in their graduation.
7. Write a JSP program to explain session handling.
8. Explain error handling in JSP with example.
9. Make a JSP page that shows a random number between 0 and 1.
10. Make an HTML form page that sends 3 parameters to a JSP page for display.
11. Make a JSP page that makes a bulleted list with a random number of entries in the list, each of which is a random int.
12. Define a method called randomInt that takes a number as an argument and returns a random int from 1 to that number. Redo exercise 3 using that method.
13. Make a JSP page that displays a random number the first time anyone visits it, and displays the same number on all subsequent visits (hint: you don뭪 need cookies or session tracking
14. Make a JSP page that always displays the same page content, but uses a background color of green, red, blue, or yellow, randomly chosen for each request. Make sure your page does not use the JSP-Styles style sheet, since that style sheet overrides the background color.
15. The java.math package has a class called BigInteger that lets you create whole numbers with an arbitrary number of digits. Create a JSP page that makes a large BigInteger from a String you supply as a request parameter, squares it, and prints out the result.
16. Make an Excel spreadsheet where each entry is a random number. Use Internet Explorer to access it if you don’t have the MS Office plugin in Netscape.Make a JSP 'signature block' page with your name and email address. Include it in another JSP page.
17. The value of the page attribute of jsp:include is allowed to be a JSP expression. Make a JSP page that includes a 'good news' page or a 'bad news' page at random.
18. You must change the mechanism used by the **Customer** class to use Java serialization to write an object out to disk and to read it back in from disk (instead of the file writing code you created yourself to save a customer to disk).. It should use the same naming convention but have the ".ser" extension instead of the ".txt" we used before.

1.Write a JSP to output the values returned by System.getProperty for various system properties such as java.version, java.home, os.name, user.name,user.home, user.dir

2.Write a JSP to output the entire line, “Hello! The time is now …” but use a scriptlet for the complete string, including the HTML tags.ie display current server time with greeting message

3.Write a JSP to output all the values returned by System.getProperties with ”

” embedded after each property name and value. Do not output the ”

” using the “out” variable.

4.Write a JSP to do either a forward or an include, depending upon a boolean variable.

5.Write a JSP/HTML set that allows a user to enter the name of a system property, and then displays the value returned by System.getProperty for that property name (handle errors appripriately).

6.Make a JSP page that randomly selects a background color for each request. Just choose at random among a small set of predefined colors.

7.Make a JSP page that lets the user supply a request parameter indicating the back-ground color. If no parameter is supplied, a background color should be selected at random

8.Write a JSP page using “jsp:forward” to go to a servlet program which display your name, date of birth and address.

9.Make a JSP page that lets the user supply a request parameter indicating the back-ground color. If no parameter is supplied, the most recently used background color (from a previous request by any user) should be used.

10.The java.math package has a class called BigInteger that lets you create whole numbers with an arbitrary number of digits.

Create a JSP page that makes a large BigInteger from a String you supply as a request parameter, squares it, and prints out the result. Use the online API at http://java.sun.com/j2se/1.5.0/docs/api/ to see the syntex for the BigInteger constructor and squaring operations.

11.Make an HTML “signature” block with your name and email address. Include it in two JSP pages.

12.Suppose that you have two different JSP pages that do two different things. However, for both pages you want to let the user supply a bgColor attribute to set the background color of the page. Implement this, but use an include mechanism to avoid repeating code. For example:

White background: http://host/path/page1.jsp

White background: http://host/path/page2.jsp

Red background: http://host/path/page1.jsp?bgColor=RED

Yellow background: http://host/path/page2.jsp?bgColor=YELLOW

13.Make two separate JSP pages that have bulleted lists containing random integers in a certain range. Avoid repeating code unnecessarily by including a page that defines a randomInt method.

14.Make a “color preference” form that collects the user’s preferred foreground and background colors. Send the data to a JSP page that displays some message using those colors. This JSP page should use a default value for any form value that the user fails to supply (but don’t worry about empty strings). So,for example, if the user goes directly to the JSP page (bypassing the form), the JSP page should still work fine. For now, don’t worry about the user sending you whitespace; just handle totally missing values.

15.Write a JSP that takes the user’s name and age from a form.

-Echo backs the name and age along with a message stating the price of movie tickets.

-The price is determined by the age passed to the JSP.

-If the age is greater than 62, the movie ticket price is $7.00.

-If the user is less than 10 years old, the price is $5.00.

-For everyone else, the price is $9.50.

16.Write a JSP that will allow a user to enter two values, select a type of mathematical operation to apply against them, and then upon clicking Submit, will display the result of the operation.

An example of the default entry form:

A valid code must also have:

-one jsp only – Math.jsp – that submits to itself

-four types of operations – add, subtract, multiply, divide

-display some kind of error if the user attempts to divide by 0

17.Create a database table to store contact information, then write a web application to manage viewing, adding, and deleting contacts from that table.

The view should look similar to:

A LIST OF ALL CONTACT

ACTION ID FNAME LNAME PHONE CITY

DEL 1 AA SS 00 BPL

DEL 2 BB SS 00 BPLADD NEW CONTACT

An item to note about insert: once the user clicks submit, the form directs to a ContactsInsertServlet.java, which insert a record into the database then redirects to the view.

A description of each file needed for this exercise:a. Write a sql script that creates a Contacts table in mysql that holds text values for id, firstName, lastName, phoneNumber,city, state, and zip. Add to the script a couple of INSERT statements so that the table will not be empty. Run your script in Squirrel to confirm it works, and so you have the table on hand for the rest of the exercise. Save the script in the root of the web application as Contacts.sql.b. ContactsViewServlet.java – displays an html table containing all records in the table. Also displays links toContactsDeleteServlet.java and ContactsInsert.html.c. ContactsInsert.html – have text fields for each field in the database. On submit, execution moves to ContactsInsertServlet.java.d. ContactsInsertServlet.java – uses an INSERT statement to add a record to the database. After the INSERT has been performed, Redirect back to ContactsViewServlet.java.e. ContactsDeleteServlet.java – uses a DELETE statement to records from the database. After the DELETE has been performed,

redirect back to ContactsViewServlet.java.

f. web.xml – must have valid references to ContactsViewServlet.java, ContactsInsertServlet.java, and ContactsDeleteServlet.java.

Note: this task is planned using servlet here but it can also be done using JSP.

18.Write a jsp application that introduce session tracking in which Servlet integrates with JSPs in the style of MVC.You need three JSPs:

a. showResult.jsp [in WEB‐INF] –

a.i. display the text corresponding to the key

a.ii. form:

a.ii.1. button: name=”more”

a.ii.2. button: name=”logout”

a.ii.3. (if more, display getDate.jsp; if logout, remove session; display login.jspb. getDate.jsp [in WEB‐INF] ‐

b.i. must display the user’s name somewhere on the page

b.ii. must contain the word Date

b.iii. requires a form

b.iii.1. button: with name=”submit”

b.iii.2. textfield: name=”key”c. showResult.jsp [in WEB‐INF] –

c.i. display the text corresponding to the key

c.ii. form:

c.ii.1. button: name=”more”

c.ii.2. button: name=”logout”

c.ii.3. (if more, display getDate.jsp; if logout, remove session; display login.jsp

19.Write a program using jsp and JDBC for developing an online application for the shopping of computer science books. (Hint: use concept of session tracking) You have to create a database for book title, author(s) of book, publisher, year of publication, price. Make necessary assumptions for book shopping.

20.Create an HTML form to take customer information (Name, Address,Mobile No.). Write a JSP program to validate this information of customers.

21.Make a random number tag that inserts a random number (double) between 0 and 1 into the page. For example, you might use it like this: . Use the Javabased approach where you extend SimpleTagSupport. Recall that Math.random() returns a double between 0 and 1.

22.Make a random int tag that inserts a random integer between 1 and some optional limit (default 10). For instance:”<sample:randomInt limit=”782″ />” . If you multiple the result of Math.random() by the appropriate value, cast the result to an int, and add 1, you can easily generate random integers from 1 to some limit.

23.Make a tag that results in whatever text it encloses being displayed in a large, bold, red,blinking format. For example:<sample:annoying>This is a test</sample:annoying>.

24.Make a JSP page that creates a list of 13 random numbers. Use a simple JSP expression for each individual random number:<%= Math.random() %>.

25.Create a servlet that stores a bean containing an array of first names (simple Strings),then forwards to a JSP page using RequestDispatcher. In the destination JSP page, loop down the array and put them into a bulleted list.

26.Create a servlet that makes an array of Name objects, which have firstName and lastName properties. Have your JSP page make an HTML table with first names in the left table cell and last names in the right table cell. Use the JSP 2.0 expression language as well as JSTL.

27.Make a JSP that displays a bulleted list of last names of all employees.

28.Make a JSP that computes the the sum of the salaries of all employees.

29.Make an HTML form that collects a last name. Send the name to a servlet or JSP page. If there is an employee with that last name, show full details on him or her (just show the first employee if there are multiple people with the same name). If there is no employee with that last name, say so.