CSHT-503

Topic	Reference	Weightage	Chapter	Sections / Page Nos.	Remarks
Javascript	[1]	17	8, 9, 10		
Java	[2]	18	1	1.6, 1.8	Exclude Advanced Topics, Random Facts, Quality Tips
			3	3.1, 3.2, 3.4, 3.5, 3.7, 3.8	
			4	4.1 to 4.7	
			5	5.1 to 5.4	
			6	6.1 to 6.3	
			7	7.1 to 7.7	
			8	8.1, 8.3, 8.6 to 8.9	
			9	9.1 to 9.3	
			10	10.1 to 10.8	
			11	11.1 to 11.6	
Java Beans	[3]	10	29		
JDBC	[4]	10	6	124-151	
JSP	[5]	20	1 to 4, 6 to 9		

References:

[1] Web Enabled Commercial Application Development Using Html, Dhtml, javascript, Perl, CGI, By Ivan Byross, BPB Publications, 2009

[2] Big Java, Cay Horstmann, Wiley Publication, 3rd edition, 2009

[3] Java 7, The Complete Reference, Herbert Schildt, 8th Edition, 2009

[4] The Complete Reference, J2EE, TMH, Jim Keogh, 2002

[5] Java Server Pages, Hans Bergsten, Third Edition, O' Reilly Media

CSHP – 503

JavaScript

1. Create a student registration form. Create functions to perform the following checks:
   a. Roll number is a 7 digit numeric value
   b. Name should be an alphabetical value (String)
   c. Non-empty fields like DOB
2. Implement a Static Password Protection.
3. Write a java script to sort an array using bubble sort. Take the number of elements and array from user.
4. Write a JS to implement stack methods (push and pop).
5. Write a JS
   a. to change the colour of text using setTimeOut()
   b. to move an image across screen using setInterval()

Java

1. Implement Matrix class. Use methods for addition, subtraction, multiplication and transpose. Use toString() to display the matrix.
2. Create a package 'Arithmetic' containing two classes 'Rational' and 'Complex'.
   a. Rational performs arithmetic with rational nos. Use integer variables to represent the private instance variables of the class, numerator and denominator. Write a constructor method that enables an object of this class. Use toString() method.
   b. Complex performs arithmetic for complex numbers. Use instance variables: real and imaginary for storing complex numbers. Simulate addition, subtraction, multiplication of complex numbers. Introduce 'this' keyword.
3. Simulate a BankAccount. Variables: accountNumber, balance. Methods: deposit, withdraw, getBalance, taxDeduction etc... Use enum for gender. And a final variable for taxRate. Implement checks such as more money cannot be withdrawn than balance. In case this happens, an Exception must be thrown. Try using javadoc.
4. Toggle nth bit in an integer.
5. Implement an Abstract Class Stack with methods push, pop, display for two classes: StaticStack and DyanamicStack. StaticStack uses one dimensional integer array to store numbers and DyanamicStack uses an integer ArrayList to store.
6. Create a base class called Shape. It should contain 2 methods getcoord() and showcoord () to accept X and Y coordinates and to display the same respectively. Create a sub class called Rect. It should also contain a method to display the length and breadth of the rectangle called showCoord(). In main method, execute the showCoord() method of the Rect class by applying the dynamic method dispatch concept.
7. WAP that copies one file to another. Pass the names of the files through command-line arguments.

JDBC

1. Create a table 'Student' in 'College' database and insert two rows in this newly created table using JDBC API.

2. Update an already created table 'Teacher' in 'College' database by updating a teacher's name, with "Dr." appended before the name, whose name is "Rita".

3. Repeat the same thing for all the teachers using PreparedStatement.

4. Delete the student with ID=3 from 'Student' database.

5. Insert two students to the ResultSet returned by the query which selects all students with FirstName="Ayush". The database must also get updated along with ResultSet.

6. Create a procedure in MySQL to count the number of Rows in table 'Student'. Use CallableStatement to call this method from Java code.

JSP Practical list

1.Display the pattern:

1

1 2

1 2 3

Take 'n' in a textbox from user. Display this pattern using

- Scriptlets
- <c:forEach> loop

2. Make two files as follows:
   a. main.html: shows 2 text boxes and 3 radio buttons with values "addition", "subtraction" and "multiplication"
   b. operate.jsp: depending on what the user selects perform the corresponding function (Give two implementations: using request.getParameter() and using expression language)

3. Validate User input entered in a form. The input must include Name, DOB, Email ID, Lucky Number, Favorite food etc. (Refer Chapter 8)
4. Display Good Morning <uname>, Good Afternoon <uname> or Good Evening <uname> based on the current time of the day.
5. Let the user enter a word a in a textbox and let her/him select one of even or odd radio buttons. If she/he selects odd, check the odd positions in the word entered, if they all contain vowels, then display the message 'You win', else display 'You lose'. Similarly, if

the user selects even, check for vowels in all even positions in the word entered. Use jstl's 'fn' library.

6. Create your custom library which contains two tags: <hello>, <choco>.
   Usage of the tags:
   - <hello name="Ajay">: Output should be Hello Ajay. It contains a mandatory attribute 'name' which can accept Dynamic value.
   - <choco texture="Chewy">: Output should be FiveStar, BarOne.
     <choco texture="Crunchy">: Output should be Munch, KitKat.
     That means the mandatory attribute must accept a value, and based on the attributes value, it should give output. You must use a bean ChocoBean for this purpose.
7. Create a custom tag "substring" with 3 mandatory attributes "input", "start", "end" which will do substring operation on given input
8. Create a custom tag "reverse" with a mandatory attribute "input "to reverse a string.
9. Create a custom tag "today" that displays today's date and time
10. Ask a user's name and age on a HTML form. Then display Hello <uname> on a JSP. On the same page ask the product the user would like to buy. Then redirect to another JSP which would display: Hello <uname>, You have ordered <product>. (Use Session Scope Variable using setTag)