Internet Technologies Guidelines
B.Sc. (H) Computer Science 5th Semester

S. No.	Topics	Reference	Contents	# of Lectures	Marks Weightage
1	Java	[1]	Ch. 2 (2.1 - 2.7); Ch. 7 (7.1 - 7.7)	(5)	(7)
2	Java Script	[2]	Ch. 8,9,10	(12)	(18)
3	JDBC	[3]	Ch. 6	(10)	(10)
4	JSP	[4]	Ch. 1,2,3,4 (upto page no. 38), 5,6,7,8, 9 (Self Study), 10 (upto pg. no. 134) before sharing session and application data, 12 (upto page 195), 21 (page 421-426)	(25)	(30)
5	Java Beans	[5]	Ch. 29	(8)	(10)
		[4]	Ch. 20 (Pg. 410-412) Value Beans. Also discuss student Bean example and Employee Bean example.		

References:

[1] Big Java by Cay Horstman - 3rd Edition
[2] Web Enabled Commercial Application Development using HTML, JS, DHTML & PHP by Ivan Bayross - 4th Edition
[3] J2EE- The Complete Reference by Jim Keogh
[4] Java Server Pages by Hans Bergsten - 3rd Edition
[5] Java 7 - The Complte reference by Herbert Schildt - 8th Edition

Internet Technologies Practical Guidelines
B.Sc. (H) Computer Science 5th Semester

Java

1. Implement a Bank Account having Instance variables: Account Number, Balance and having methods:

   float Deposit (float x)
   float withdraw (float x)
   int get account no ()
   float get balance ()
   tax deduction ()

   Then implement class Bank having an array list of accounts of type BankAccount. Implement following methods:

   AddAccount in Bank
   Get Total balance in Bank
   Get account number with max. and min. balance
   Find an account given a bank account no.
   Count no. of accounts having atleast specific balance

2. 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.

JavaBeans

1. Implement Student JavaBean using Serializability Interface.
2. Implement Employee JavaBean using Serializability Interface.

JDBC

1. Create Student and Results Database and perform the following using JDBC programs
   a. Find total number of students
   b. Print average marks for each subject input by user.
   c. Find the name of student getting highest marks.
   d. Find no of students getting first, second and third division.
   e. Find subject wise toppers
   f. Find the average marks
   g. Find the student getting second highest marks.

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

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. DOB entered in dd/mm/yy format and should be display in words (e.g. Saturday, January 01, 2000)
   d. Check on non-empty fields
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 JavaScript to implement stack methods (push and pop).

5. Write a JavaScript
   a. to change the color of text using setTimeOut()
   b. to move an image across screen using setInterval()
6. Implement the question no. 1 of hands on exercises of chapter 10 (page 190).

HANDS ON EXERCISES

1. Create a web page, which accepts user information and user comments on the web site. Design the web page using form elements and check if all the Text fields have being entered with data else display an alert. Obtain an output as shown in the diagrams 10.11.1, 10.11.2 and 10.11.3.

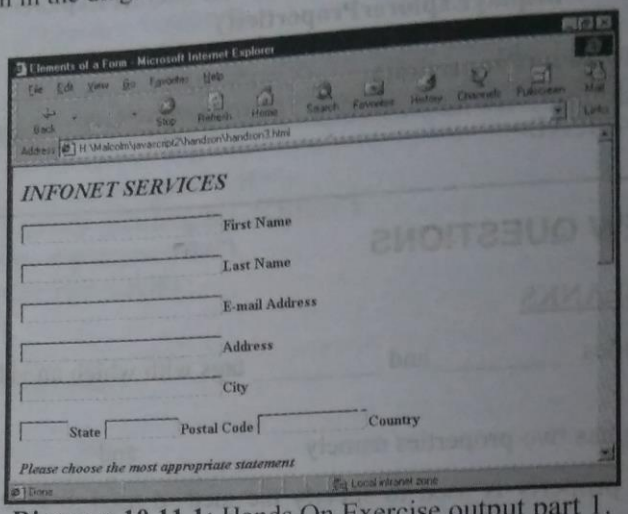

Diagram 10.11.1: Hands On Exercise output part 1.

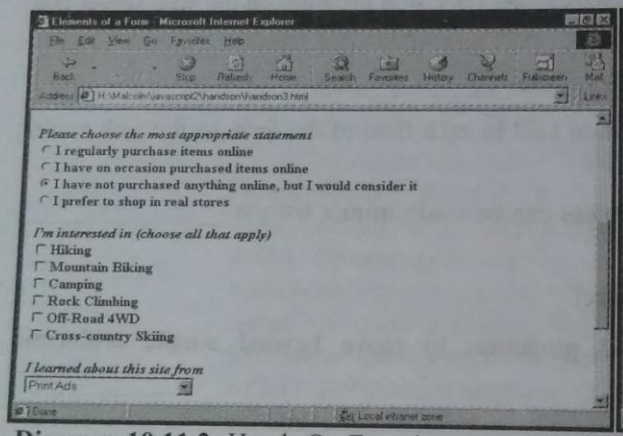

Diagram 10.11.2: Hands On Exercise output part 2.

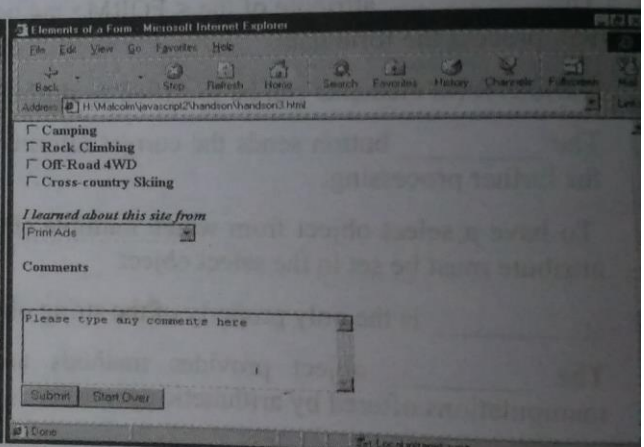

Diagram 10.11.3: Hands On Exercise output part

JSP

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)