

Java and J2EE Lab

Course Code: ISL67

Credit: 0:0:1

Contact Hours: 28

Course Content

1. Write a Program that simulates a telephone that records missed incoming calls. For each missed call, store the time of call, telephone number of origin, and name of the caller if the name is available. For unlisted numbers, set the name to "private caller". Choose or extend the most appropriate collection class and provide the following features.

Numbers are recalled in the order they arrive

Up to 10 numbers are recorded. When the eleventh call comes in, it is stored and the oldest call is deleted so that no more than 10 numbers are ever recorded.

After each number display, the user can select

- i. To delete the call
- ii. To go on to the next missed call, or
- iii. To display the call details (number, caller name and time).

Delete the number if user specifies a number to delete.

Write a helper class to represent an incoming call with fields to hold the number, name of the caller, and time of the call. Write a tester call that stores the several numbers, simulate the user pressing the missed-calls button, and finally prints the entire collection of stored calls.

2. Write a Java program using user-defined storage classes to create a book database and store it in a Collection List. Books collection should include title, author, publisher and price. Also write a method to sort the books in ascending order of price and store it in another List. Maintain the book details with respect to a unique book id. Prompt for an author name and list all the books with the same author name. Create a new list holding all the book details with price greater than a user specified price. For a given a value by the user, find all the books that match either the whole or a part of the book title. Identify a publisher and print books from a particular publisher. Update the publisher details based on a title.

3. Create a desktop java application using swings to enable an user to enter student information such as name, usn, age, address, sgpa of 8 semesters, category. Perform validations on all the fields. Display appropriate messages in pop up boxes to indicate wrong entries. On clicking of the "compute" button, find the cgpa. On clicking of the "done" button, mouse place the student details in a collection. Display the collection in a textarea on the click of a button.

4. Write a java program using swings to validate user login information using dialog boxes. Once validated, allow the user to enter the customer id, if the person is a new customer, else check whether the customer exists in a collection and obtain the

<p>customer id. The customer id can be obtained given a mobile number. Allow the user to enter the item purchased by giving the item id and quantity purchased. On clicking of a button, the item name and the total cost should appear in the corresponding GUI components. Using option dialog box, indicate the types of discount available for the customer. On clicking on the print button, print the details in information dialog box.</p>
<p>5. Write a program that uses Java Swing and JDBC to create a stand-alone application:</p> <p>Create two tables namely, Representative (<u>RepNo</u>, RepName, State, Comission, Rate) and Customer (<u>CustNo</u>, CustName, State, Credit_Limit, <u>RepNo</u>) in MySQL database. Use appropriate Swing components to insert values in a form. Use another form to display Representative's information when Credit_Limit is above 15,000.</p>
<p>6. Create a Servlet to file IT returns that accepts personal information, salary information and Tax deduction details from the user and write the information into a file. Also accept the name of the person and display in on the page.</p>
<p>7. Create a servlet that accepts patient information in a hospital such as patient id, patient name, age, date of admission, cause of admission, doctor diagnosed, treatment proposed. Place the details into a database. Allow options to insert , update ,view and delete the contents in the database.</p>
<p>8. Write a JSP and Servlet Program to do the following to buy a T-Shirt online:</p> <ol style="list-style-type: none"> A set of checkboxes to select your T-Shirt accessories such as 'belt', 'cap', 'hair-band' etc. A text area / text field to enter your T-Shirt tag-line A Radio-button that allows the user to choose between T-Shirt with chest pocket and without. A Combo Box to choose your T-Shirt color Appropriate labels for these GUI Components A Button called "Click Me" which when pressed will Insert the details entered into a table called 'TShirts'. An OrderNo is generated by adding '1' to the existing 'OrderNo' If 'TShirts' table is empty the initial value of 'OrderNo' is 100. This 'OrderNo' is also inserted into the 'TShirts' table Display all the records of the 'TShirts' table in tabular form <p>PS: Frontend display should be in JSP and the business logic should be written in Servlet Class.</p>
<p>9. Create a Telephone Directory Application using Servlet that searches the database based on phone number or name. Also show database table creation with inserting 2-3 values to the table.</p> <ol style="list-style-type: none"> Database Name: OnlineDirectory

b. Table Design:

Table Name: Telephone_Directory

Attributes: Phone_Number, Name, Address, Company, Pin_Code.

10. Write a program using JSP that helps a student to calculate the income tax on various annual incomes that he will be earning when he gets a job.

a. *Login.html* will call *dataCapture.jsp* that should do the following:

- i. Use Java Collections to make a list of valid users and facilitate user login functionality.
- ii. Give a personalized Welcome message and display today's date.
- iii. Have a Text Entry with label 'Name' to enter the name of the user.
- iv. Give a List of Organizations to choose 'Place of Work'
- v. Provide a Male or Female option to choose the 'Gender'
- vi. Have a Text Entry with label 'Annual Income'
- vii. Give a Submit button reading 'Calculate Tax'

b. *CalculateTax.jsp* must calculate the interest based on the following business rules:

- i. Salary below 1,00,000 shall no have income-tax.
- ii. Calculate 15% of tax on 1,00,001 – 5,00,000.
- iii. Calculate 20% on 5,00,001 onwards.

PS: The final income tax along with the details of how it is calculated must be put in a session object and displayed to the user in *dataCapture.jsp*. All the income taxes calculated so far by the user, must be taken out of the session object and displayed, each time in *dataCapture.jsp* which has a link called 'Logout' that destroys the session.

11. Create two tables Flight (Flight_Number, Airline_Name, Weekdays) and seatReservation(Flight_Number, Date, Seat_Number, Customer_Name, Customer_Phone) in MySQL database.

Create JSP page *ReserveOnline.jsp* to reserve an airline seat and insert the values into the table SeatReservation. OnClick of Submit in *ViewDetails.jsp* display information about reservation. Validate the Flight_Number from already existing Flight database and generate random number for Seat_Number within the range 1-500.

Also create a link to display information of all the flights running on a particular day.

12. a. Write a Java Program that creates two threads object of Thread class. Where one thread asks the user to enter a number not less than four digits. Split the digits of the number and display in words the value of the number. Ex: 1 – One. Second thread finding the number of vowels in a word. Ex: JAVA – Vowel - A, Count – 2.

b. Write a program using java.net to transfer file from client socket to server socket.

Text Books:

1. Herbert Schildt, 'The Complete Reference Java (J2SE 5 Edition)', TATA McGRAW-HILL

Edition 2005.

2. Ivan Bayross, Sharanam Shah, Cyntiha Bayross and Vishali Shah, 'Java EE 5 for Beginners', SPD (Sharoff Publishers & Distributors Pvt. Ltd.), 2nd edition August 2008.

Reference:

1. Jim Keogh, 'The Complete Reference J2EE', TATA McGRAW-HILL Edition 2002.
2. B V Kumar, S Sangeetha, S V Subrahmanya, J2EE Architecture, TATA McGRAW-HILL Edition 2007.

Course Outcomes:

At the end of the course, students will be able to

1.	Develop programs using collections such as lists, hashmaps and sets. (PO-1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3))
2.	Develop front ends using swings and manipulate stored data. (PO-1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3))
3.	Develop web applications using servlets, jsp and manipulate the data. (PO- 1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3))
4.	Perform concurrent execution of programs to improve CPU utilization. (PO-1(2), 2(2), 3(3),5(3)) (PSO-1(3),2(3))
5.	Develop reports and communicate effectively on the concepts related to Java and J2EE. (PO-1(1),2(2), 10(3)) (PSO-2(2))