Course Code & Name	26CAPL8001 Java Programming Lab		
Teaching Hours /week (L: T:P: R)	0:0:4:0	Credits	2
CIE Marks	50	ESE Marks	50
Course Type	Lab (PCL)	Exam Hours	3 Hrs.
Prerequisites (if any)	26CAPC8001		

Course Objectives:

1. To help students implement multithreading and client-server communication in Java using the Runnable interface and Socket programming.
2. To train students to develop interactive web applications using Servlets and JSP with session management, request handling, and MVC concepts.
3. To enable students to perform JDBC-based database connectivity, CRUD operations, transaction management, and advanced queries.
4. To introduce students to enterprise application development using the Spring Framework, focusing on IoC, DI, Spring MVC, and Spring JDBC/Boot.

Course Outcomes (COs)

At the end of the course students should be able to:

CO#	Course Outcome	Bloom's Knowledge Level (KL)
CO 1	Students will be able to implement multithreading and client–server communication using the Runnable interface, threads, sockets, and networking APIs.	K3
CO 2	Students will be able to develop interactive web applications using Servlets and JSP with session handling, request processing, and MVC-based design	K4
CO 3	Students will be able to perform database connectivity, CRUD operations, transaction management, and advanced SQL queries using JDBC.	K3

CO 4	Students will be able to build enterprise-level applications using the Spring Framework, applying IoC, DI, Spring MVC, and Spring JDBC/Spring Boot features.	**K4**

Note: *K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO 1	3	2	2	3	2	-	-	2
CO 2	3	2	3	2	2	1	1	2
CO 3	3	3	2	3	2	-	-	2
CO 4	3	2	3	2	3	1	1	3

Note: *1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), - : No Correlation*

Course Assessment Method

(CIE: 50 marks, ESE: 50 marks)

Continuous Internal Evaluation Marks (CIE)

Attendance	Continuous Assessment (Lab Experiments/Project, Viva, Completion of Lab Reports / Project Report)	Internal Lab Examination-1	Internal Lab Examination- 2	Total
8	**22**	**10**	**10**	**50 Marks**

End Semester Examination Marks (ESE)

Submission of Record: Students shall be allowed for the end semester examination only upon submitting the duly certified record.

Procedure/ Algorithm/ Flowchart	Conduct of experiment/ Execution of work	Quality of Output	Viva voce	Record	Total
10	**15**	**10**	**10**	**5**	**50 Marks**

Continuous Assessment (22 Marks)

1. Conduct of Lab Experiments/Project (12 Marks)
2. Timely Completion of Lab Reports/ Project Reports (5 Marks)
3. Viva Voce (5 Marks)

Final Marks Averaging: The final marks for conduct of experiments, viva, and records are the average of all the specified experiments in the syllabus.

Evaluation Pattern for End Semester Examination (50 Marks)

1. Procedure/Algorithm/Flowchart (10 Marks)
2. Conduct of Experiment/Execution of Work (15 Marks)
3. Quality of Output (10 Marks)
4. Viva Voce (10 Marks)
5. Record (5 Marks)

Exp. No.	List of Experiments
1	Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).
2	Write a Java program to demonstrate client-server communication using Sockets based on the TCP/IP protocol. Implement a Server program that waits for client connections, receives a message from the client, and sends an appropriate response back. Implement a Client program that connects to the server, sends a message, and displays the response received from the server. Ensure proper handling of input/output streams and closing of connections after communication.
3	Develop a simple web application to demonstrate the **Servlet lifecycle**. Create a class LifecycleServlet that overrides init(), service(), and destroy() methods to display messages in the browser indicating when each method is executed. Deploy and test it on Apache Tomcat.
4	Design a web application that collects user details such as name, age, and email through an HTML form. Implement a Servlet that handles this form submission —

	use the doGet() method to display a personalized welcome message and the doPost() method to insert the submitted data into a database using JDBC.
5	Create a web application that demonstrates different **session management techniques** in Servlets. In the first version, maintain session data using Cookies. In the second version, use URL Rewriting to track users across requests. In the third version, use the HttpSession interface to store and retrieve user information such as username or login time.
6	Develop a web application that implements **Request Dispatching, Filters, and Event Listeners**. • The LoginServlet should validate user credentials and forward valid requests to WelcomeServlet using RequestDispatcher. • Create a **Filter** to log each request before processing. • Implement a **ServletContextListener** to initialize database parameters when the application starts and release them on shutdown.
7	Create a JSP page to demonstrate the use of **JSP scripting elements** — declarations, scriptlets, and expressions. Display your name, the current date, and the result of a simple calculation (like factorial or square of a number) using these scripting elements.
8	Develop a JSP page that integrates with a database using JDBC. • Display all student records from a student table in tabular format. • Include an HTML form on the same page that allows adding a new student record. • Use JSP **implicit objects** (request, response, application, out) to handle data submission and display appropriate messages.
9	Create a JSP page to display user profile data using **Expression Language (EL)** and **JSTL (JSP Standard Tag Library)**. Retrieve user data (like name, email, and date of registration) from a JavaBean or database and display it using EL. Use JSTL tags to iterate through lists, format numbers or dates, and conditionally display messages if no user data is available.

10	Write a Java program to establish a connection between a Java application and a MySQL database using JDBC. The program should load the JDBC driver, connect to the database, and display an appropriate message indicating whether the connection is successful or failed. Include proper exception handling and resource closure.
11	Design and implement a Java program using JDBC Statement to perform all four CRUD operations (Insert, Update, Delete, and Retrieve) on a table named student(id, name, course, marks). Allow the user to choose the operation through console input and display the results accordingly.
12	Write a Java program using PreparedStatement to insert multiple student records into a database. Demonstrate how using parameterized queries with PreparedStatement prevents SQL injection attacks. Display a success message after inserting each record.
13	Create a stored procedure in your MySQL database named getDeptSalary(dept_id) that returns the total salary of all employees belonging to a specific department. Write a Java program using CallableStatement to call this procedure, pass the department ID as an input parameter, and display the result returned from the procedure.
14	Develop a Java program that retrieves all records from an employee table using JDBC. Use ResultSet to fetch the data and ResultSetMetaData to dynamically display the column names and corresponding values without hardcoding them.
15	Write a Java program to demonstrate **transaction management** in JDBC. Create two related tables, for example orders and order_details. Perform insertion into both tables within a single transaction, and implement a rollback operation if any insertion fails, ensuring database consistency.
16	Implement a Java program to demonstrate **batch processing** in JDBC. Insert at least five product records into a product(id, name, price) table using addBatch() and executeBatch() methods. Display the number of records successfully inserted as the output.
17	Modify one of your previous JDBC programs to demonstrate **robust exception handling** using try-catch-finally and try-with-resources. Ensure that database

	resources like Connection, Statement, and ResultSet are properly closed even in the event of an error.
18	Develop a **Spring Boot web application** that performs basic database operations using **Spring MVC** and **Spring JDBC**. • Create a controller to handle web requests and map them to appropriate service methods. • Connect the application to a database using JDBC. • Display a list of employee records on a webpage using Thymeleaf or JSP. • Provide options to add and delete employees through a simple user interface. • Test the application by running it as a Spring Boot project.

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Servlet, JSP and Spring MVC: A Tutorial	Budi Kurniawan & Paul Deck	Brainy Software Inc.	2014
2	Spring in Action	Craig Walls	Manning Publications	7th Edition, 2023.

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	ava: The Complete Reference	Herbert Schildt	McGraw Hill Education	12th Edition, 2022
2	JDBC 4.2, Servlet 3.1, and JSP 2.3 Includes JSF 2.2 and Design Patterns – Black Book	Santosh Kumar K	Dreamtech Press / Wiley India	2nd Edition

3	Spring Framework with Java: A Practical Approach	B. Mohamed Ibrahim	Laxmi Publications Pvt Ltd	1st Edition Jan 2016

Web References/ Video Links (NPTEL, SWAYAM…)	
Sl.No	**Link ID**
1	NPTEL – Internet Technology (Client-Server Programming in Java) https://archive.nptel.ac.in/courses/106/105/106105084/
2	YouTube – Java Web Application | Java Servlets | JSP | JDBC | MySQL https://www.youtube.com/watch?v=wN8pcb8kM3A
3	NPTEL – Internet Technology (Client-Server Programming in Java) https://archive.nptel.ac.in/courses/106/105/106105084/
4	SWAYAM – Web Technology https://onlinecourses.swayam2.ac.in/nou25_cs09/preview
5	NPTEL – Enterprise System using Java Spring Framework https://archive.nptel.ac.in/courses/106/105/106105214/
6	YouTube – Spring Framework Full Course | Spring Core, MVC, Boot https://www.youtube.com/watch?v=If1Lw4pLLEo

FISAT
Focus on Excellence