Course: MCA **Semester:** 4

Prerequisite: Knowledge of Core Java Programming and HTML.

Rationale: To make the students aware of the data driven web applications, web services, MVC architecture and ORM concepts of modern web application development.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
4	0	4	-	6	20	20	20	60	30	150

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Java Database Connectivity** Introduction to JDBC, JDBC architecture. Common JDBC Components - DriverManager, Driver, Connection, Statement, ResultSet, SQLException. Types of JDBC drivers, Establishing connection with database, Submitting queries and obtaining results, ResultSetMetadata Interface, Precompiled statements, CallableStatement, JDBC transactions.	15	8
2	**Java Servlets** Configuring Java Development Environment, Downloading and configuring Apache Tomcat web server. Introduction to servlets, Servlet life cycle, Exploring servlet API, Basic servlet structure, Handling client requests, Generating response, Working with cookies, Session tracking. Servlet filters - Introduction to filters, creating custom filters, mapping filters to servlets, mapping filters to specific URL pattern.	20	10
3	**Java Server Pages** Introduction to JSP, Need of JSP, Advantages of JSP over servlet, Life cycle of a JSP page, JSP basic tags: declaration tag, scriptlet tag, expression tag. JSP implicit objects - out, request, response, application, session, pageContext, page, exception. Introduction to java beans, Reading and writing properties of JavaBeans in JSP pages. JSP 2.0 expression language - Use of EL, Invoking Expression Language, Preventing Expression language evaluation, Accessing scoped variables. Custom tag libraries - tag handlers, tag library descriptor. JSP standard tag library (JSTL): Downloading and installing JSTL in Java web application, Core tags (c:out, c:forEach, c:if, c:choose, c:set, c:remove, c:import, c:catch).	25	12
4	**Java Web Services** Introduction to web services, Architecture of web services, Types of web services - SOAP web services, REST web services SOAP v/s REST, Introduction to REST, Configuring RESTful web service using Jersey framework in java web application, RESTful architectural principles, HTTP method and URI matching. JAX-RS injection - PathParam, QueryParam, FormParam. Introduction to Server responses and exception handling, JAX-RS client API.	20	12
5	**Introduction to Spring and Hibernate** Introduction to Spring MVC Framework, Spring MVC Architecture and Components, Advantages of Spring MVC, Modules Of Spring Framework. Spring Application Introduction to Hibernate, Architecture of hibernate, Hibernate O/R mapping, Configuring hibernate development environment, Implementing hibernate O/R Mapping, Introduction to Hibernate Query Language (HQL).	20	12

Reference Books

1.	**Java Server Programming Java EE6 Black Book (TextBook)** Dreamtech Press
2.	**RESTful Java with JAX-RS 2.0** By Bill Burke | O'Reilly | 2nd Edition
3.	**Core Servlets and Java Server Pages Volume-1** By Mary Hall and Larry Brown | Prentice Hall | 2nd Edition
4.	**Core Servlets and Java Server Pages Volume-2** By Marty Hall, Larry Brown and Yaakov Chaikin | Prentice Hall | 2nd Edition

Course Outcome

After Learning the Course the students shall be able to:

1. design and develop data driven web applications using JDBC, servlet and JSP API.
2. develop MVC based web applications using Struts.
3. develop and integrate web services and web clients.
4. describe significance of Object Relational Mapping (ORM) in Java context.
5. construct ORM-based web applications using Hibernate.

List of Practical

1.	**JDBC Connectivity** Develop a java application using concept of JDBC for user login. User will enter username and password. Application will match for the same in database table. If match is found then display message "Successful Login" and if not found then display message "Invalid Username and password ".
2.	**JDBC Connectivity & CRUD Operations** Develop a Menu driven java application for student information which will create the table with appropriate columns. Menu will contain the options like insert, update and delete. Based on the option, data will be inserted or updated or deleted from table based on student id (student_id will be primary key). Display appropriate message for each operation.
3.	**Data Driven GUI Application** Develop an application using GUI for user registration. In first form user will enter the data for registration (e.g. Name, Birthdate, Email_id, Phone_No). When user clicks on submit button, data will be inserted into database and user will be redirected to another form. In this form data will be selected from database and displayed in proper format.
4.	**A Data Driven Servlet Application** Develop a data driven servlet application for user authentication. Also display header values of request object and response object.
5.	**Servlet - Session Management** Develop a servlet application for implementing student grading system. In this system on first page student will enter his data (name, id and department). By clicking on next button he will be redirected to another page. On the second page he will enter marks of all 6 subjects. By clicking on "Generate result" he will be redirected to next page. On this page all information of student will be displayed along with the result (pass or fail) in appropriate color. Also display the name of student on each page he visits. (Use session management)
6.	**Servlet - Request Redirection**

	Develop an application for the following: User should be redirected to the URL entered in location text box.
7.	**JSP - User Authentication** Develop a data driven JSP application for user authentication.
8.	**JSP - Java Beans** Develop an application for Bank Account information using Java Beans. This application will perform various operation on bank account like withdraw, deposit and displaying the information.
9.	**JSP - Custom Tags** Develop a custom JSP tag which accepts 10 numbers from user and sorts them in specified order.
10.	**JSP - Application Context** Develop an application that will allow maximum 3 users to access the application. If number of users exceeds than limit then user will be redirected to the error page.
11.	**Simple RESTful Service** Develop a simple JAX-RS service that provides currency conversion.
12.	**Data Driven RESTful Service** Develop a RESTful service to accept student registration data sent from a web form.
13.	**RESTful Web Service - CRUD Operations** Develop a JAX-RS service that provides CRUD database operation interface for a phonebook. Call appropriate service from servlet/jsp pages to perform data oriented operations.
14.	**JAX-RS Client** Develop a JAX-RS client that consumes RESTful service developed in Program-11. Utilize the client in UI layer (JSP pages).
15.	**JAX-RS Client** Develop a RESTful service to perform authentication of the user. The JAX-RS client will be utillized in servlet and provide interface between jsp page and web service to perform authentication.
16.	**Struts2 - User Authentication** Develop a struts 2 application that accepts user name and password from user. If both values are valid then user will be redirected to home page otherwise redirected on error page.
17.	**Struts2 - User Registration and Authentication** Develop a data driven Struts 2 application that accepts registration details from user, i.e. name, contact number, address, user id and password. After successful registration, application redirects user to log in page. If user provides valid

	authentication details then application will redirect user to home page and will display welcome message, otherwise user will be redirected to error page.
18.	**Struts2 - Data Driven Application** Develop a phone book using struts 2. Application accepts data of contacts like name, address, contact number and email id. The application should provide features to update, delete and list data of contacts into/from database.
19.	**Struts2 - Interceptors** Create a logging service using struts 2 to monitor log in and log out timestamp of the users. (Hint: utilize interceptors)
20.	**Struts2 - Hibernate Integration** Develop an application using struts 2 that provides insert, update, delete and list operations for Student data like, name, roll number, semester, course, result (%). The data oriented operations must be performed through hibernate.