Java Web Application Development with Servlets, JSP

1. **Introduction to Web Applications:**
   - **Objective**: Understand the fundamentals of web applications and how they function in a client-server architecture.
   - **Topics**:
     - Definition of web applications.
     - Key components: web server, application server, and database.
     - HTTP protocol and its role in web communication.
2. **HTML and Form Handling:**
   - **Objective**: Learn the basics of HTML and how to create forms for user input.
   - **Topics**:
     - HTML structure and elements.
     - Creating HTML forms.
     - Input elements (text fields, radio buttons, checkboxes).
     - Form submission using the POST and GET methods.
3. **Servlets:**
   - **Objective**: Understand Servlets and how to handle HTTP requests and responses.
   - **Topics**:
     - Servlets as Java classes.
     - Servlet life cycle: init, service, destroy.
     - Configuring Servlets in web.xml.
     - Handling different HTTP methods (GET, POST).
4. **JSP (JavaServer Pages):**
   - **Objective**: Introduce dynamic content generation using JSP.
   - **Topics**:
     - JSP syntax and scripting elements.
     - JSP directives (page, include).
     - Using expressions, scriptlets, and declarations in JSP.
5. **JSTL (JavaServer Pages Standard Tag Library):**
   - **Objective**: Enhance JSP development with reusable tags for common tasks.
   - **Topics**:
     - Core JSTL tags (c:if, c:forEach, c:set, etc.).
     - Formatting and localization with fmt tags.
     - Handling collections and arrays in JSTL.
6. **JDBC (Java Database Connectivity):**

- **Objective**: Enable interaction with databases in web applications.
- **Topics**:
  - Establishing database connections.
  - Executing SQL queries and updates.
  - Handling result sets and database errors.
  - Prepared statements and connection pooling.

7. **Session Management:**

- **Objective**: Manage user sessions to store and retrieve data.
- **Topics**:
  - Understanding HTTP sessions.
  - HttpSession object and its methods.
  - Cookies and URL rewriting for session tracking.

8. **Error Handling and Exception Handling:**

- **Objective**: Handle errors and exceptions gracefully in web applications.
- **Topics**:
  - Configuring error pages in web.xml.
  - Handling exceptions using try-catch blocks.
  - Custom error pages and error codes.

9. **WAR Files (Web Application Archive):**

- **Objective**: Package and deploy web applications in a standard format.
- **Topics**:
  - Creating a WAR file structure.
  - Deployment descriptor (web.xml).
  - Deploying a WAR file to a servlet container like Tomcat.

10. **Servlet Filters:**

- **Objective**: Understand request and response filtering for various purposes.
- **Topics**:
  - Filter interface and filter chains.
  - Implementing filters for authentication, logging, and data transformation.
  - Filter configuration in web.xml.

11. **Authentication and Security:**

- **Objective**: Secure your web application with user authentication and access control.
- **Topics**:
  - Implementing basic authentication with servlets.
  - Role-based access control (user roles and permissions).
  - Best practices for secure coding.

12. **RESTful Web Services (Optional):**
    - **Objective**: Explore RESTful services using Servlets for API development.
    - **Topics**:
        - Introduction to REST architecture.
        - Building RESTful APIs with Servlets.
        - Consuming RESTful services in Java web applications.
13. **MVC (Model-View-Controller) Architecture:**
    - **Objective**: Learn the MVC pattern for structured web application development.
    - **Topics**:
        - MVC fundamentals: Model, View, Controller.
        - Separation of concerns in web applications.
        - Advantages of MVC.
14. **Deployment to Tomcat:**
    - **Objective**: Understand the deployment process on a Tomcat server.
    - **Topics**:
        - Installing and configuring Apache Tomcat.
        - Deploying web applications to Tomcat.
        - Server configuration (e.g., server.xml and context.xml).
15. **Project Work:**
    - **Objective**: Apply the learned concepts to a practical project.
    - **Outcomes**: Students should complete a web application project using Servlets and JSP, demonstrating their understanding of web development concepts.

Outcome: By covering these topics and objectives, students will gain a solid foundation in Java web application development using Servlets, JSP, JSTL, JDBC, WAR files, and Tomcat deployment, enabling them to build dynamic and secure web applications.