**Assignment Questions 7**

💡 **Q1.What is the use of JDBC in java?**

JDBC (Java Database Connectivity) is a Java API that allows Java programs to interact with databases. It provides a set of classes and methods to connect to a database, execute SQL queries, retrieve and manipulate data, and manage database transactions. JDBC enables Java applications to access various relational databases in a standardized and database vendor-independent manner.

💡 **Q2.What are the steps involved in JDBC?**

The steps involved in JDBC are:

1. Import JDBC Packages: Import the required JDBC packages in your Java program.

2. Load the JDBC Driver: Load the appropriate JDBC driver class using `Class.forName()`.

3. Establish a Connection: Use `DriverManager.getConnection()` to establish a connection to the database.

4. Create a Statement: Create a statement object to execute SQL queries.

5. Execute the Query: Execute the SQL query using the statement object.

6. Process the Results: Process the results returned by the query.

7. Close the Connection: Close the database connection and release resources.

💡 **Q3.What are the types of statement in JDBC in java?**

1. Statement: Basic SQL statement used for executing queries. It has limited functionality and is vulnerable to SQL injection.

2. PreparedStatement: Precompiled SQL statement with placeholders for parameters, offering better performance and security.

3. CallableStatement: Used to call stored procedures in the database, allowing both input and output parameters. It's suitable for complex database interactions.

💡 **Q4.What is Servlet in Java?**

Servlet in Java is a server-side technology that extends the capabilities of a web server. It receives and processes HTTP requests from clients (e.g., web browsers) and generates dynamic content in response. Servlets are primarily used to develop web applications, handling tasks such as form data processing, session management, and database interactions.

💡 **Q5.Explain the life Cycle of servlet?**

The lifecycle of a servlet consists of several stages:

1. Initialization: The servlet is loaded and initialized by the container.

2. Service: The container calls the `service()` method to handle client requests.

3. Request Handling: The `service()` method dispatches the request to appropriate methods (e.g., `doGet()` or `doPost()`).

4. Destruction: The servlet is destroyed by the container when it's no longer needed.

Each stage involves specific methods that developers can override to implement custom functionality.

💡 **Q6.Explain the difference between the RequestDispatcher.forward() and HttpServletResponse.sendRedirect() methods?**

RequestDispatcher.forward():

- Forwards the request from one servlet to another resource on the server-side.

- The URL in the browser remains unchanged.

- Client is unaware of the server-side forward.

- Allows passing data between servlets using request attributes.

- Useful for internal forwarding within the application.

HttpServletResponse.sendRedirect():

- Redirects the client's browser to a new URL or location.

- The URL in the browser changes to the new URL.

- Creates a new request-response cycle.

- Data cannot be directly passed between servlets, but query parameters can be used.

- Useful for external redirects to different resources or external websites.

💡 **Q7.What is the purpose of the doGet() and doPost() methods in a servlet?**

The purpose of the `doGet()` and `doPost()` methods in a servlet:

1. `doGet()`:

- Handles HTTP GET requests from clients (e.g., web browsers).

- Used to retrieve data or perform read-only operations.

- Data is sent in the URL parameters.

- Commonly used for fetching information and displaying content.

2. `doPost()`:

- Handles HTTP POST requests from clients.

- Used for submitting data to the server, often for updates or database operations.

- Data is sent in the request body, making it suitable for sensitive or larger data.

💡 **Q8.Explain the JSP Model-View-Controller (MVC) architecture.**

The JSP Model-View-Controller (MVC) architecture is a design pattern used in web development to separate application concerns:

1. Model: Represents the application's data and business logic.

2. View: Represents the presentation layer responsible for rendering the user interface.

3. Controller: Manages user input, processes requests, and updates the model and view accordingly.

MVC promotes modularity, reusability, and easier maintenance of web applications.

💡 **Q9.What are some of the advantages of Servlets?**

Advantages of Servlets:

1. Platform Independence: Servlets are written in Java, making them platform-independent and easily portable across different systems.

2. Performance: Servlets are efficient and provide better performance than traditional CGI scripts due to their ability to reuse resources.

3. Scalability: Servlets can handle multiple requests concurrently, making them suitable for scalable web applications.

4. Extensibility: Developers can extend the functionality of servlets through Java classes and libraries.

5. Integration: Servlets seamlessly integrate with other Java technologies like JSP, JDBC, and frameworks like Spring.

💡 **Q10.What are the limitations of JSP?**

LimiLimitations of JSP:

1. Complexity: JSP can lead to complex code mixing presentation and logic, making maintenance challenging.

2. Steeper Learning Curve: Developing JSP applications may require a steeper learning curve for beginners.

3. Performance: JSP pages may experience reduced performance due to the overhead of translation and compilation.

4. Limited Reusability: Reusing JSP code across different projects can be cumbersome.

5. Mixing Presentation and Logic: It may lead to code readability issues and hinder separation of concerns.