Q1.What is the use of JDBC in java?

Ans: JDBC stands for Java Database Connectivity. It is a Java API that allows Java programs to connect to and manipulate databases. JDBC is a standard API, so it can be used to connect to a variety of databases, including MySQL, Oracle, and PostgreSQL.

JDBC provides a set of classes and interfaces that can be used to perform database operations, such as:

- Connection: This class represents a connection to a database.
- Statement: This class represents a statement that can be executed against a database.
- ResultSet: This class represents the results of a query.

Q2.What are the steps involved in JDBC?

1. Ans: Loading the JDBC driver: The first step is to load the JDBC driver for the database that you want to connect to. This can be done by using the `Class.forName()` method.
2. Establishing a connection: Once the JDBC driver is loaded, you can establish a connection to the database. This can be done by using the `DriverManager.getConnection()` method.
3. Creating a statement: Once you have a connection to the database, you can create a statement. A statement is a representation of a SQL statement that can be executed against the database.
4. Executing a query: You can use a statement to execute a query against the database. A query is a request for data from the database.
5. Processing the results: Once a query is executed, you can process the results. The results of a query are represented by a `ResultSet` object.
6. Closing the connection: Once you are finished with the connection, you should close it. This can be done by using the `Connection.close()` method.

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

Ans:

- Statement: This is the most basic type of statement in JDBC. It can be used to execute any type of SQL statement, including select, insert, update, and delete statements.
- PreparedStatement: This type of statement is used to execute prepared statements. Prepared statements are pre-compiled SQL statements that can be executed multiple times with different values. This can improve performance, as the statement does not need to be compiled each time it is executed.
- CallableStatement: This type of statement is used to execute stored procedures. Stored procedures are functions or procedures that are stored in the database. They can be used to perform complex operations that would be difficult or time-consuming to implement in Java.

Q4.What is Servlet in Java?

Ans: A servlet is a Java programming language class that is used to extend the capabilities of a web server. Servlets are used to create dynamic web pages. They can be used to process requests from clients, generate responses, and interact with databases.

Servlets are typically deployed on web servers, such as Apache Tomcat or Jetty. When a client makes a request to a web server, the web server will instantiate a servlet and call its `service()` method. The `service()` method will handle the request from the client and generate a response.

Servlets are a powerful tool that can be used to create dynamic web pages.

Q5.Explain the life Cycle of servlet?

1. Ans: Instantiation: A servlet is instantiated by the web container when it receives a request for the servlet's URL.
2. Initialization: The servlet's `init()` method is called by the web container during the initialization phase. The `init()` method is used to initialize the servlet's state.
3. Service: The servlet's `service()` method is called by the web container when it receives a request for the servlet's URL. The `service()` method is used to handle the request from the client and generate a response.
4. Request processing: The servlet's `service()` method typically delegates the request processing to one of the following methods:
    - `doGet()`: This method is called when the client makes a GET request.
    - `doPost()`: This method is called when the client makes a POST request.
    - `doPut()`: This method is called when the client makes a PUT request.
    - `doDelete()`: This method is called when the client makes a DELETE request.
5. Destruction: The servlet's `destroy()` method is called by the web container when the servlet is destroyed. The `destroy()` method is used to clean up the servlet's state.
6. Pooling: In some cases, the web container may pool servlets. This means that the web container may keep a pool of servlets that are ready to handle requests.

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

Ans: The MVC architecture is a software design pattern that separates the presentation layer from the business logic layer. This separation of concerns makes it easier to develop, maintain, and test web applications.

In the JSP MVC architecture, the model represents the data that is used by the application. The view represents the presentation of the data to the user. The controller is responsible for handling the user's requests and updating the model and view as needed.

Q9.What are some of the advantages of Servlets?

- Ans: Portability: Servlets are portable, meaning that they can be deployed on any web server that supports the servlet API.
- Efficiency: Servlets are efficient, as they are designed to handle multiple requests simultaneously.
- Scalability: Servlets are scalable, meaning that they can be used to handle large numbers of requests.
- Reusability: Servlets can be reused, which can save development time and effort.

- Security: Servlets can be secured using a variety of mechanisms, such as authentication and authorization.
- Support: Servlets are well-supported by the Java community, and there are a number of resources available to help developers learn and use servlets.

Q10.What are the limitations of JSP?

- Ans: Separation of concerns: JSP pages mix presentation logic and business logic, which can make it difficult to maintain and test applications.
- Performance: JSP pages can be slow to render, especially if they contain a lot of dynamic content.
- Security: JSP pages can be vulnerable to security attacks, such as cross-site scripting (XSS).
- Complexity: JSP pages can be complex to develop, especially if they need to interact with a database or other back-end systems.