

Assignment Questions 7

Q1.What is the use of JDBC in java?

ans:-The JDBC API is a Java API that can access any kind of tabular data, especially data stored in a relational database.

JDBC helps you to write Java applications that manage these three programming activities: Connect to a data source, like a database.

Send queries and update statements to the database.

Q2.What are the steps involved in JDBC?

1. **ans:-** The fundamental steps involved in the process of connecting to a database and executing a query consist of the following:
 - i. Import JDBC packages
 - ii. Load and register the JDBC driver.
 - iii. Open a connection to the database.
 - iv. Create a statement object to perform a query.
 - v. Execute the statement object and return a query resultset.
 - vi. Process the resultset.
 - vii. Close the resultset and statement objects.
 - viii. Close the connection.

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

ans:-

The statement interface is used to create SQL basic statements in Java it provides methods to execute queries with the database. There are different types of statements that are used in JDBC as follows:

- i. Create Statement
- ii. Prepared Statement
- iii. Callable Statement

1. Create a Statement: From the connection interface, you can create the object for this interface. It is generally used for general-purpose access to databases and is useful while using static SQL statements at runtime.

Syntax:

```
Statement statement = connection.createStatement();
```

2. Prepared Statement represents a recompiled SQL statement, that can be executed many times. This accepts parameterized SQL queries. In this, “?” is used instead of the parameter, one can pass the parameter dynamically by using the methods of PREPARED STATEMENT at run time.

3. Callable Statement are stored procedures which are a group of statements that we compile in the database for some task, they are beneficial when we are dealing with multiple tables with complex scenario & rather than sending multiple queries to the database, we can send the required data to the stored procedure & lower the logic executed in the database server itself. The Callable Statement interface provided by JDBC API helps in executing stored procedures.

Syntax: To prepare a CallableStatement

```
CallableStatement cstmt = con.prepareCall("{call Procedure_name(?, ?)}");
```

Implementation: Once the callable statement object is created execute() is used to perform the execution of the statement.

Q4.What is Servlet in Java?

ans:-A servlet is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model.

Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers.

Q5.Explain the life Cycle of servlet?

ans:-The web container maintains the life cycle of a servlet instance. Let's see the life cycle of the servlet:

1. Servlet class is loaded.
2. Servlet instance is created.
3. init method is invoked.
4. service method is invoked.
5. destroy method is invoked.

1) Servlet class is loaded

The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.

2) Servlet instance is created

The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.

3) init method is invoked

The web container calls the init method only once after creating the servlet instance. The init method is used to initialize the servlet. It is the life cycle method of the javax.servlet.Servlet interface. Syntax of the init method is given below:

```
public void init(ServletConfig config) throws ServletException
```

4) service method is invoked

The web container calls the service method each time when request for the servlet is received. If servlet is not initialized, it follows the first three steps as described above then calls the service method. If servlet is initialized, it calls the service method. Notice that servlet is initialized only once. The syntax of the service method of the Servlet interface is given below:

```
public void service(ServletRequest request, ServletResponse response)  
    throws ServletException, IOException
```

5) destroy method is invoked

The web container calls the destroy method before removing the servlet instance from the service. It gives the servlet an opportunity to clean up any resource for example memory, thread etc. The syntax of the destroy method of the Servlet interface is given below:

```
public void destroy()
```

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

ans:- RequestDispatcher - forward() method

- 1.When we use the forward method, the request is transferred to another resource within the same server for further processing.
- 2.In the case of forward, the web container handles all processing internally and the client or browser is not involved.
- 3.When forward is called on the requestDispatcherobject, we pass the request and response objects, so our old request object is present on the new resource which is going to process our request.

4. Visually, we are not able to see the forwarded address, it is transparent.
5. Using the `forward()` method is faster than `sendRedirect`.
6. When we redirect using `forward`, and we want to use the same data in a new resource, we can use `request.setAttribute()` as we have a request object available.

SendRedirect

1. In case of `sendRedirect`, the request is transferred to another resource, to a different domain, or to a different server for further processing.
2. When you use `sendRedirect`, the container transfers the request to the client or browser, so the URL given inside the `sendRedirect` method is visible as a new request to the client.
3. In case of `sendRedirect` call, the old request and response objects are lost because it's treated as new request by the browser.
4. In the address bar, we are able to see the new redirected address. It's not transparent.
5. `sendRedirect` is slower because one extra round trip is required, because a completely new request is created and the old request object is lost. Two browser requests are required.
6. But in `sendRedirect`, if we want to use the same data for a new resource we have to store the data in session or pass along with the URL.

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

ans:- `doGet()`

In `doGet()` Parameters not encrypted.

`doGet()` allows bookmark.

doget() method is idempotent.

In doget() not change anything on the server.

doget() is request information.

In doget() parameters are appended to URL and sent with header information.

doget() is Faster.

In doget() only 1024 characters limit.

doPost()

In dopost() Parameters encrypted.

dopost() disallows bookmark.

dopost() method does not idempotent..

In dopost() server is expected to remember.

dopost() is provide information.

In dopost(), on the other hand, will send the information through socket back to the webserver and it won't show in the URL bar.

dopost() Slower.

In dopost() does not have limit.

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

ans:-

MVC stands for Model View and Controller. It is a design pattern that separates the business logic, presentation logic and data.

Controller acts as an interface between View and Model. Controller intercepts all the incoming requests.

Model represents the state of the application i.e. data. It can also have business logic.

View represents the presentaion i.e. UI(User Interface).

Q9.What are some of the advantages of Servlets?

ans:-

A servlet can be imagined to be as an applet running on the server side. Some of the other server side technologies available are Common Gateway Interface (CGI), server side JavaScript and Active Server Pages (ASP). Advantages of servlets over these server side technologies are as follows:

- **Persistent:** Servlets remain in memory until explicitly destroyed. This helps in serving several incoming requests. Servlets establishes connection only once with the database and can handle several requests on the same database. This reduces the time and resources required to establish connection again and again with the same database. Whereas, CGI programs are removed from the memory once the request is processed and each time a new process is initiated whenever new request arrives.
- **Portable:** Since servlets are written in Java, they are portable. That is, servlets are compatible with almost all operating systems. The programs written on one operating system can be executed on other operating system.
- **Server-independent:** Servlets are compatible with any web server available today. Most of the software vendors today support servlets within their web server products. On the other hand, some of the server side technologies like server side JavaScript and ASP can run on only selected web servers. The CGI is compatible with the web server that has features to supports it.
- **Protocol-independent:** Servlets can be created to support any of the protocols like FTP commands, Telnet sessions, NNTP newsgroups, etc. It also provides extended support for the functionality of HTTP protocol.

- Extensible: Servlets being written in Java, can be extended and polymorphed into the objects that suits the user requirement.
- Secure: Since servlets are server side programs and can be invoked by web server only, they inherit all the security measures taken by the web server. They are also safe from the problems related to memory management as Java does not support the concept of pointers and perform garbage collection automatically.
- Fast: Since servlets are compiled into bytecodes, they can execute more quickly as compared to other scripting languages. The bytecode compilation feature helps servlets to give much better performance. In addition, it also provides advantage of strong error and type checking.

Q10.What are the limitations of JSP?

ans:-the limitation of JSP. As you know, the assumption behind a JSP document is that it provides a single overall presentation. But what if you want to give totally different results depending on the data that you receive?

Beans and custom tags, although extremely powerful and flexible, but they don't overcome the limitation that the JSP page defines a relatively fixed top-level page appearance. The solution is to use both servlets and Java Server Pages.

If you have a complicated application that may require several substantially different presentations, a servlet can handle the initial request, partially process the data, set up beans, then forward the results to one of a number of different JSP pages,

depending on the circumstances.