

## JDBC

1) Define jar file & explain need for jar file.

→ jar means java archive, it is used to compress many files into one single file based on zip file format.

⇒ Jar file is needed to import the properties based on the requirement.

2) Define design pattern | Explain factory design pattern

⇒ an optimized sol<sup>n</sup> for commonly re-occurring design problems is known as design pattern.

Eg mvc Architecture

⇒ factory is the one which creates / produces diff objects of same type.

⇒ factory generally takes on type of string type

3) Define the following

i) Abstraction :- Hiding the implement<sup>n</sup> & providing the functionality to the user with the help of interface is known as Abstraction.

ii) Loose coupling :- change in the implement<sup>n</sup> which do not affect the user is known as loose coupling.

iii) Interface :- Interface is a media to common b/w users & any device.

4) factory / helper method: used to create implement<sup>n</sup> object.

5) NullPointerException & pointing towards an object which is Not present throws an Exception called Null pointer Exception.

4) Why JDBC Driver is an implement<sup>n</sup> of JDBC API?

Since all the Driver classes must mandatorily implements java.SqL.Driver which is the part of JDBC API, hence JDBC Driver is an implement<sup>n</sup> of JDBC API.

5) Name Driver classes along with port Number.

Oracle - OracleDriver (1521)

mySQL - Driver (3306)

mg-SQL - SqLServerDriver (1433)

Derby - EmbeddedDriver (1527)

6) What are the specification of JDBC?

There are 3 diff specifications present for JDBC namely

1) All the Driver classes must Contain one Static block in it

2) All Driver classes must mandatorily implements java.SqL.Driver interface which is the part of JDBC API.

3) All Driver classes must mandatorily be registered with DriverManager by using a static method called registerDriver()

7) Define JDBC & mention the steps & advantages of JDBC.

→ Java Database Connectivity (JDBC) is a specification given in the form of Abstraction API to achieve loose coupling b/w java Application & DB server.

⇒ Therefore six different steps

⇒ Load & register Driver class

⇒ establish the connection with DB server

⇒ Create statement / platform

⇒ Execute the SQL query / statement

⇒ Process the resultant data

⇒ Close all costly Resources.

### Advantages

⇒ We can Achieve loose coupling b/w java App & DB

⇒ Platform independent

8) Explain Costly Resources?

Resource which uses of System properties in the form of Stream are known as costly Resources.

9) What is the outcome of DML & DQL?

→ The outcome of DML is 0-N integer value which gives the total no of records affected in the DB.

→ The outcome of DQL is processed / Resultant data which is stored in cursor / Buffer memory which can be fetched with the help of ResultSet interface which is a part of JDBC API.

10) Define the following.

1) execute(): execute is a generic method, Since it is used to execute any type SQL queries / SQL statements.

⇒ Hence, the return type of execute() is boolean & return boolean <sup>true</sup> value in case of DQL & boolean false value in case of DML / DDL query.

2) executeUpdate(): It is a specialized method. Since it is used to execute only DML query.  
⇒ the outcome of DML is 0-N integer Value hence return type is integer.

3) executeQuery(): It is a specialized method. Since it is used to execute DQL queries.

⇒ the outcome of DQL queries is processed / Resultant Data which is present in cursor / Buffer memory which can be fetched by using ResultSet Interface, which is a part of JDBC API.

⇒ hence return type is ResultSet Interface.

ii) Define place holder & Rules to set data for place holder.

⇒ place holder is a parameter which takes dynamic value at the runtime by the user.

⇒ there are 3 diff rules in order to set the data for a place holder namely

1) we have to set the data for a place holder before the execution

2) the no of data must exactly matched with no of place holders

3) we have to set the data for placeholder by using setXXX().

- (a) Which method is used to fetch the processed  
(or) Resultant data from the cursor/buffer memory

getXXX() of ResultSet interface.

- (b) Difference b/w next() & absolute()?

next()

\* Used to check whether next Record is present in the cursor/buffer memory

\* Next() is used whenever there are minimum Records present in the cursor/buffer memory

absolute()

\* Used to check particular Record is present in cursor/buffer memory (or) not based on the parameter called Integer Row Number.

\* absolute() is used when ever there are 'n' no. of records in cursor/buffer memory.

- (c) What is return type for SetXXX & getXXX()

=> the return type for getXXX() are Respective data types where return type for SetXXX() is void

- (d) Where is polymorphism exhibited in JDBC?

Polymorphism in case of JDBC is exhibited in getConnections() Since it returns implement object of Connection Interface w.r.t. to diff DB Server/vendor based on their URL's

insert into Btm. Student values (999)

stmt.setInt(1, 420);

stmt.setString(2, "Bharathi"); } o/p  
1, Bharathi, 32.45

stmt.setDouble(3, 32.45); } Overridden Value.

stmt.setInt(1, 1);

stmt.executeUpdate();

16) Can we Deal with multiple records by using Statement Interface?

yes, But it is not a good practice Since compilation takes place each time along with execution. The performance of an app decreases hence we use preparedStatement Interface

17) In General, is Statement interface faster in performance than Prepared Statement?

In General, prepared Statement interface is faster in performance for two important reasons namely

i) it supports the concept of placeholder to take dynamic values at the Runtime by the user.

ii) Since P.S.I Supports the concept of Compile once & Execute many times [Execution plan]

18) How do we establish a connection b/w Java Appln & Db Server?

⇒ By using `getConnection()`.

19) How are all the data automatically saved into the database Server

⇒ By default the autoCommit mode is set to Boolean true value. Bcz of which All the data are automatically saved into the db server.

20) When & How to disable the auto Commit mode?

⇒ to disable the autoCommit mode, we use a method called `SetAutoCommit()` & passing a boolean false argument.

We have to explicitly disable the AutoCommit mode

before beginning Txn but After establishing  
a connection with database Server.

Q1) How to explicitly save the data into the db Server  
⇒ By using `commit()` at the end of Txn

Q2) Relate JDBC Txn with atomicity & explain?

\* Atomicity means do everything (or) do nothing

\* do everything means complete / successful Txn where if all the db operations are successfully executed, then the data are considered to be valid which has to be saved into the db Server, leading to data consistency.

\* Do nothing refers to incomplete / unsuccessful Txn where if any one of the db operation fails then the rollback operation is called which is used to rollback the entire executed db operations.

Txn starts from the beginning without saving any data into the db Server due to data inconsistency.

Q3) Short Note on JDBC Txn.

Def<sup>n</sup>: JDBC Txn is a single business unit, which may have multiple SQL statements which must be executed.

- \* JDBC Txn is needed to maintain data Consistency in the db Server.
- \* the advantage of JDBC Txn is to achieve ACID properties/ ACID Rules. where
  - A - Atomicity
  - C - Consistency
  - I - Isolation
  - D - Durability

Q4) Why Prepared Stmt are also known as Pre-Compiled Stmt.

⇒ Since the query is passed at the time of implementation object creation of prepared Statement interface but not in the execute(), hence prepared Stmt are also known as pre-compiled Stmt.

## Servlet

1) Define Server & types of Server & Example

⇒ Server is a software which manages all the resources along with which process the client Request & Serve the Client Request.

The diff types of Servers are

1) Database Server

2) Application Server

3) Web Server.

Database Server: used to deal with data.

example oracle, my-SQL, ms-SQL, derby, mongo Db etc.

Application Server: used to execute a Dynamic

Real time Appl'

Ex: ej J-boss, IBM-websphere & oracle-web logic

Webserver: used to execute only web Applications.

example Apache tomcat Server, oracle GlassFish

2) Define Web-apps & work folder of Apache-tomcat Server.

-er.

⇒ Web-apps :- it is used to deploy all the web-applications onto a Apache tomcat Server

work-folder: it is used to store the data w.r.t. to the Translated Servlet [conversion of Jsp to Servlet]

3) Define Deployment & mention the port no for Apache tomcat Server?

⇒ making all the Resources Available to the Server is known as deployment.

⇒ By default, the port no for Apache tomcat Server is 8080.

- => By the default welcome file / landing page is index
- => the default root tag for an xml file is <web-app>
- => whenever the type of Request is not mentioned / configured, then the default type of request is get Request.

\*\*

4) How to carry the data in a Secured manner to the Server.

=> By using post Request that means In case of Post Request, the data are carried to the Server as a part of http request body which is not displayed even to the end user [client]. hence the data are secured.

\*\*

5) How are all the Configurations made in web.xml understood by the JEE container?

All the custom / user defined tags are transformed (or) converted into 8 bit unicode format based on which all the configurations made in web.xml are understood by the JEE container.

6) Who loads all applications within the server?  
JEE Container load which is an engine that is used to manage all the JEE Components such as Servlet, Jsp, EJB etc...

7) Define Servlet & Explain types of Servlet in Brief  
Servlet is a Server side java program of which performs all the three diff types of logic such as presentation logic, persistence logic & business logic along with which process the http client Request

index  
web-apps  
is  
to  
ie of  
er  
- dis  
uto  
xml  
atted  
sed  
xml  
19  
st  
such  
f  
ich  
as  
zic

get back some http Response.  
There are two diff types of Servlet present namely  
1) GenericServlet  
2) HttpServlet

### GenericServlet

- \* it is not specific to any protocol
- \* it does not Support Session.
- \* it contains both Abstract & Concrete methods in it
- \* If it is an abstract class present javax.Servlet package
- \* It is mandatory to override Service() Since it is an abstract method & also Responsible for processing the Client Request

### HttpServlet

- \* it is Specific to http protocol.
- \* it Supports Session.
- \* it Contains only concrete methods
- \* It is an abstract class present in javax.Servlet.http package
- \* We have to override respective Concrete methods called doXXX() for a particular type of http request

Such When ever we override Service() or doXXX(), it throws an Exception called ServletException & IOException.

8) What are the properties to be configured for a Servlet in web.xml

Each & every Servlet has to be configured with 3 diff properties in Web.xml namely

- 1) ServletName [unique]
- 2) Url-pattern [unique]
- 3) FQCN

9)

10) How to make Servlet as Single threaded?

→ By default Servlet is multithreaded, but it can be made as Single threaded in two ways, namely

- 1) by writing a Servlet class which implements a marker interface called SingleThreadedModel. Which is a deprecated interface [outdated]
- 2) by making Service() as synchronized.

11) Define UI/form data?

The data which is entered by the end user [client] on a form page & is submitted to the server in the form of key & value pair is known as UI/form data.

12) Which method is used to fetch UI/form data?

The UI/form data can be fetch by using a method called getParameter()

⇒ In this method, the key is taken as an argument. If the key is present, then method returns associated value.

⇒ If the key is not present then it returns null but not any error/Exception.

### 13) Define Enumeration

Enumeration is a Group of fixed data.

\* 14) Is there a ~~Service()~~ service() present in HttpServlet.

⇒ service() in case of HttpServlet is present as a Concrete method which is not a good practice to be overridden. Since HttpServlet depends upon the type of Http request.

### 15) Mention the diff Status Message & status code.

| Status Code | Status message  |
|-------------|-----------------|
| 404         | Client error    |
| 500         | Server error    |
| 200         | Success message |

### 16) Explain Servlet lifecycle in brief.

Servlet lifecycle depicts/Represents the events/phases which takes place from Servlet Object creation until Servlet object destruction.

- Servlet Life cycle will always begin only whenever Servlet object is created.

⇒ a Servlet object can always be created in two different ways namely

i) Whenever a client makes a 1st Request to a Servlet, only one Servlet object is created by the JEE Container by calling the default constructor of Servlet.

ii) In case of load-on-startup, where the JEE container creates a Servlet object by calling default Constructor of Servlet at the time of Server Start-up without waiting for the 1st client request so that the delay time made by the 1st client Request can be avoided which increases the performance of an app!

⇒ There are 4 diff lifecycle phases present for a Servlet namely

- i) Instantiation/object creation phase
- ii) Initialization phase
- iii) Service phase
- iv) destruction phase.

17) What is the value Configured for load-on-startup

⇒ load-on-startup must mandatorily be configured with a positive integer value but the JEE Container gives the priority based on the lowest +ve integer value.

\* 18) What is the o/p whenever load-on-startup is configured with a -ve integer value?

In this case the JEE Container creates a Servlet object based on 1st client request without throwing any error/Exception.

## 19) What are the situations of destroy()

destroy() is called by the JEE container to close all the costly resources only once but in two diff situations namely

1) Whenever we close an Application destroy() is called to close all the costly Resource w.r.t to that appln.

2) Whenever we re-deploy an appln on to a Server, destroy() is called to close the previously used costly Resources.

## 20) Diff b/w ServletConfig & ServletContext

### ServletConfig

- \* it is an interface present in javax.servlet package
- \* an implement<sup>n</sup> object of ServletConfig interface is created by the JEE Container by calling a factory/helper method called getServletConfig() immediately after the Servlet Object creation.
- \* only one implement<sup>n</sup> object of ServletConfig interface is created for that particular Servlet Object

- \* the data present in Servlet Config are init / initialization parameters, which is made available only to that particular Servlet object.

### ServletContext

- \* It is an interface present in javax.servlet package
- \* an implement<sup>n</sup> object of Servlet Context interface is created by JEE Container by calling a factory/helper method called getServletContext() at the time of application loading

- \* only one implement<sup>n</sup> object of Servlet Context interface is created for the one entire application

- \* The data present in Context object is Context parameters & attributes, which is made available throughout the Appln

## ServletConfig

\* the Scope of ServletConfig is limited to that particular Servlet Object

\* init / initialization parameter is a data in the form of key & value pair which is used to initialize the Resources of particular Servlet Object

\* init parameters can be fetched by using `getInitParameter()` along with the reference of Config Object.

`config.getInitParameter()`

\* Scope of ServletContent is throughout the Appln

\* Context parameters are the data in the form of Key & value pair which is made available throughout the application

\* The Context parameters can present in Servlet Content can be fetched by using `getInitParameter()` along with the reference of context object  
`context.getInitParameter()`

Q) Define Attribute, programmatic & declarative data.

⇒ Attribute is the data in the form of key & Value pair where key must be a unique String & the value can be any object

⇒ The data present in Request, Session & Content object is known as programmatic data, Since the scope of all these objects are throughout the appln

⇒ The data which is declared on an XML file is known as declarative data.

22) Explain Servlet Chaining in brief.

⇒ chaining from one Servlet to Another Resource which can either be html, Jsp (or) Another Servlet is known as Servlet chaining.

⇒ Servlet Chaining is needed to make the data present in one Servlet Available to Another Resources which can either be html, Jsp (or) Another Servlet.

⇒ Servlet Chaining will always begin whenever a client makes any Request to a Servlet.

⇒ Servlet Chaining can always be performed in two diff ways namely

i) Request Dispatcher

ii) Send Redirect.

---

1) code for ui / form Data

2) Code for Servlet life cycle.

3) Code for Saving data's entered on the web page dynamically into the db Server in a Secured manner? [post]

4) code for get Request