1. What is JDBC??

>> JDBC stands for **J**ava **D**ata**b**ase **C**onnectivity, which is a standard Java API for database-independent connectivity between the Java programming language

Common Tasks of JDBC:

- Making a connection to a database.
- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & Modifying the resulting records.

2. JBDC Package??

   >> java.sql and javax.sql

3. JDBC Drivers?

   >> It is and interface that interacts with database server. There are 4 Types of JDBC Drivers

Type-1: JDBC-ODBC Bridge Driver

Type 1 driver, a JDBC bridge is used to access ODBC drivers installed on each client machine**.**

TYPE-2: JDBC-Native API

In a Type 2 driver, JDBC API calls are converted into native C/C++ API calls, which are unique to the database

TYPE-3 JDBC- NET PURE JAVA

In a Type 3 driver, a three-tier approach is used to access databases. The JDBC clients use standard network sockets to communicate with a middleware application server. The socket information is then translated by the middleware application server

TYPE-4 100% Pure Java

In a Type 4 driver, a pure Java-based driver communicates directly with the vendor's database through socket connection.

4. **What are the steps to use JDBC API?**

   >>

   1. Import **JDBC** packages.
   2. Load and register the **JDBC** driver.
   3. Open a connection to the database.
   4. Create a statement object to perform a query.
   5. Execute the statement object and return a query resultset.
   6. Process the resultset.
   7. Close the resultset and statement objects.
   8. Close the connection.

5. Loading driver class?

   >> **forName("**driver**.** class**");** loads **the specified JDBC** driver**. When the** driver loads**, it also registers itself with the DriverManager . Hence, when** you **call DriverManager#getConnection()** you**'re able to establish the Connection through the** driver loaded**.**

6. **What are the requirements to create connection?**

   Register the driver

   Class.forName(driver-name);

   Get Connection

   DriverManager.getConnection(url,username,password)

7. What are the Statements?

>> 3 Statements: Statements(general purpose database access) , PreparedStatements(accept input parameter at runtime), CallableStatements(to access stored procedures)

8. Statements and PreparedStatements

SELECT * FROM USER WHERE ID=1;

String sql = SELECT * FROM USER WHERE ID=?;

PreparedStatement ps = connection.PrepareStatemnet(sql);

ps.setString(1,id);

ps.executeQuery();

9. **What is ResultSet?**

A ResultSet **object is a table of data representing a database** result set**, which is usually generated by executing a statement that queries the database.**

10. **Methods of ResultSet?**

public boolean next(): ...
public boolean previous(): ...
public boolean first(): ...
public boolean last(): ...
public boolean absolute(int row): ...
public boolean relative(int row): ...
public int getInt(int columnIndex): ...
public int getInt(String columnName):
public int getString(int column index)
public string getString(String columnName)

11. What is Transaction?

Transaction represents **a single unit of work**.

The ACID properties describes the transaction management well. ACID stands for Atomicity, Consistency, isolation and durability.

Atomicity means either all successful or none.

Consistency ensures bringing the database from one consistent state to another consistent state.

Isolation ensures that transaction is isolated from other transaction.

Durability means once a transaction has been committed, it will remain so, even in the event of errors, power loss etc.

12.Scrollable ResultSet

>> **A scrollable ResultSet is one which allows us to retrieve the data in forward direction as well as backward direction but no updations are allowed**

13.Execute executeUpdate executeQuery

execute **method can be used with any type of SQL statements and it returns a boolean.**

executeQuery **method execute statements that returns a result set by fetching some data from the database. It executes only select statements.**

executeUpdate **method execute sql statements that insert/update/delete data at the database.**

Servlet

14.What is servlet?

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.**

15.What is web.xml? What are filters

The web. xml file **is the standard deployment descriptor for the** Web **application that the** Web **service is a part of. It declares the filters and servlets used by the service.**

Servlet Filters are Java classes that can be used in Servlet Programming for the following purposes −

- To intercept requests from a client before they access a resource at back end.
- To manipulate responses from server before they are sent back to the client.

There are various types of filters suggested by the specifications −

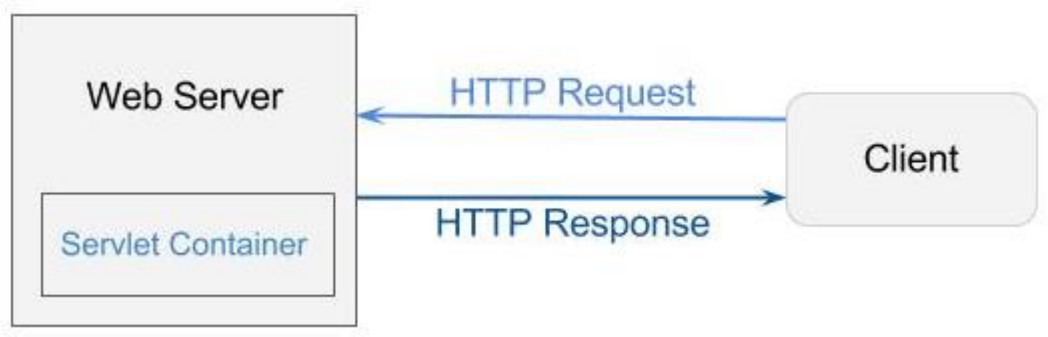
- Authentication Filters.
- Data compression Filters.
- Encryption Filters.
- Filters that trigger resource access events.
- Image Conversion Filters.
- Logging and Auditing Filters.
- MIME-TYPE Chain Filters.
- Tokenizing Filters .
- XSL/T Filters That Transform XML Content.

Filters are deployed in the deployment descriptor file **web.xml** and then map to either servlet names or URL patterns in your application's deployment descriptor.

16. Servlet Container?

It provides the runtime environment for JavaEE (j2ee) applications. The client/user can request only a static WebPages from the server. If the user wants to read the web pages as per input then the servlet container is used in java.

The servlet container is the part of web server which can be run in a separate process. We can classify the servlet container states in three types:



Servlet Container States

The servlet container is the part of web server which can be run in a separate process. We can classify the servlet container states in three types:

- **Standalone:** It is typical Java-based servers in which the servlet container and the web servers are the integral part of a single program. For example:- Tomcat running by itself
- **In-process:** It is separated from the web server, because a different program runs within the address space of the main server as a plug-in. For example:- Tomcat running inside the JBoss.
- **Out-of-process:** The web server and servlet container are different programs which are run in a different process. For performing the communications between them, web server uses the plug-in provided by the servlet container.

The Servlet Container performs many operations that are given below:

- Life Cycle Management
- Multithreaded support
- Object Pooling
- Security etc.

17.Servlet Life Cycle

The servlet is initialized by calling the **init()** method.

The servlet calls **service()** method to process a client's request.

service() method has doGet() and doPost() method

The servlet is terminated by calling the **destroy()** method.

18. How to write text response using Servlet?

    We set the content type of the **response** object to **text**/plain . ServletOutputStream sout = **response**. getOutputStream(); We get the ServletOutputStream which is used to send character **text** to the client.

19. RequestDispacther and Redirect

    The **RequestDispatcher** interface allows you to do a server side forward/include whereas **sendRedirect()** does a client side **redirect**. **SendRedirect()** will search the content **between** the servers.

20. **Forward and include**

    The RequestDispatcher **interface allows you to do a server side forward/include whereas** sendRedirect() **does a client side** redirect**.** SendRedirect() **will search the content** between **the servers.**

21. **Session**

    The time interval in which two systems(i.e. the client and the server) communicate with each other can be termed as a session**. In simpler terms, a** session **is a state consisting of several requests and response between the client and the server.**

22. Session Tracking Techniques

Cookies

Hidden Form Field

URL Rewriting

HttpSession

23.**Cookies and Session Difference**

Cookies and Sessions are used to store information. **Cookies** are only stored on the client-side machine, while sessions get stored on the client as well as a server.

Cookie	Session
• Cookies are client-side files that contain user information	• Sessions are server-side files which contain user information
• Cookie ends depending on the lifetime you set for it	• A session ends when a user closes his browser
• You don't need to start cookie as it is stored in your local machine	• In PHP, before using $_SESSION, you have to write session_start(); Likewise for other languages
• The official maximum cookie size is 4KB	• Within-session you can store as much data as you like. The only limits you can reach is the maximum memory a script can consume at one time, which is 128MB by default
• A cookie is not dependent on Session	• A session is dependent on Cookie
• There is no function named unsetcookie()	• Session_destroy(); is used to destroy all registered data or to unset some

24.What are servletScope?

Application

Session

Request
Page

25. Context Parameter

Context parameters are additional value pairs that are sent to your app in the request URL from the host application. Using **context parameters**, your apps can selectively alter their behavior based on information provided by the application

Servlet Context is basically referred to as an object which has information regarding application and the Web Container. With Servlet context we can log events, get the URL of the specific resource, and can easily store the attributes for other servlets to use.
The core advantage of Servlet is that it is easy to maintain and acts as a mediator between the container and servlet.

There are some important methods of servlet context which are given below:
getInitParameter () – return the value of parameter.
getInitParameterNames () – returns the name of parameter.
void setAttribute () – used to set the values of attributes.
void getAttribute () – used to get the values of attributes.
void removeAttribute () – used to remove the attribute.

26. Singleton class

The **singleton** design **pattern** is used to restrict the instantiation of a **class** and ensures that only one instance of the **class** exists in the JVM. In other words, a **singleton class** is a **class** that can have only one object (an instance of the **class**) at a time per JVM instance.

27. How can you break Singleton pattern in Java?

Deserialization. In serialization, we can save the object of a byte stream into a file or send over a network. Suppose if you serialize the **Singleton class**, and then again de-serialize that object, it will create a new instance, hence deserialization will **break** the **Singleton pattern**.

28.JSP Tags

Scriplet tags: <% %> Java Code

Declaration tags: <%! %> Declare methods and varaibles used in code

Expression tags: <%= %> Expressions values of varirables

- Directives tags: page—This directive lets you configure an entire JSP page, such as whether its output is HTML or XML. You'll see more on this directive in Day 6.
- include—This directive lets you include another page or resource in a JSP page.
- taglib—This directive lets you use a set of custom JSP tags as defined in a tag library. More on taglibs is coming up in Day 9, "Using Custom JSP Tags," and Day 10, Creating Custom Tags."

29.Jsp Standard Action Tags

- We can dynamically insert a file, reuse the beans components, forward user to another page, etc. through JSP Actions like include and forward.

- jsp:useBean
- jsp:include
- jsp:setProperty
- jsp:getProperty
- jsp:forward
- jsp:plugin
- jsp:attribute
- jsp:body
- jsp:text
- jsp:param
- jsp:attribute
- jsp:output

30.ORM

Object relational mapping — or **ORM**, is a design pattern for converting (wrapping) that data stored within a relational database into an object that can be used within an object oriented language

31. Hibernate

Hibernate **is an Object-Relational Mapping (ORM) solution for JAVA.** Hibernate **maps Java classes to database tables and from Java data types to SQL data types**

32. What is JPA persistence?

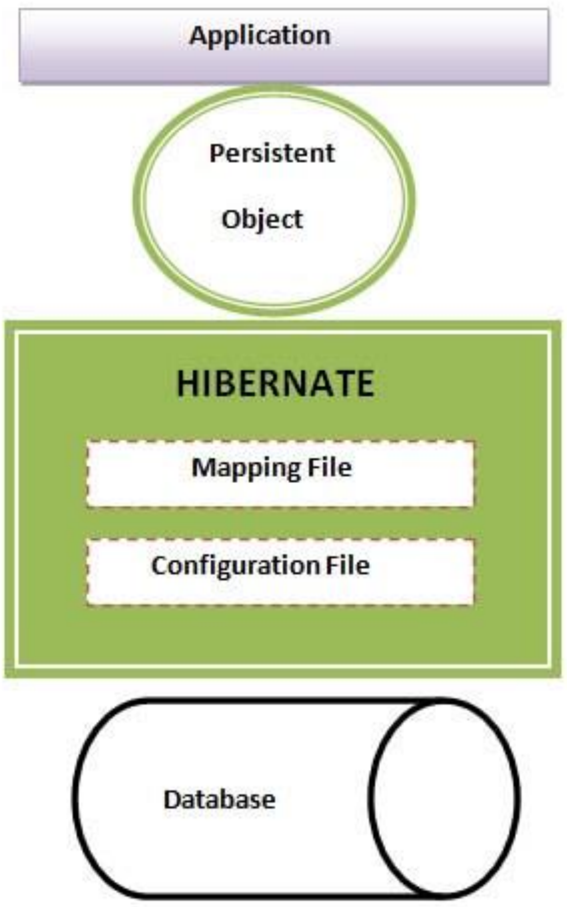
Persistence simply means to Store Permanently. In JAVA we work with Objects and try to store Object's values into database(RDBMS mostly). **JPA** provides implementation for Object Relation Mapping(ORM) ,so that we can directly store Object into Database as a new Tuple.

33. Hibernate Architecture

The Hibernate architecture is categorized in four layers.

- Java application layer
- Hibernate framework layer
- Backhand api layer
- Database layer

Let's see the diagram of hibernate architecture:



SessionFactory

The SessionFactory is a factory of session and client of ConnectionProvider. It holds second level cache (optional) of data. The org.hibernate.SessionFactory interface provides factory method to get the object of Session.

Session

The session object provides an interface between the application and data stored in the database. It is a short-lived object and wraps the JDBC connection. It is factory of Transaction, Query and Criteria. It holds a first-level cache (mandatory) of data. The org.hibernate.Session interface provides methods to insert, update and delete the object. It also provides factory methods for Transaction, Query and Criteria.

Transaction

The transaction object specifies the atomic unit of work. It is optional. The org.hibernate.Transaction interface provides methods for transaction management.

ConnectionProvider

It is a factory of JDBC connections. It abstracts the application from DriverManager or DataSource. It is optional.

TransactionFactory

It is a factory of Transaction. It is optional.

34. What is hibernate cfg file?

    Hibernate Configuration File(**cfg file**) is the **file** loaded into an **hibernate** application when working with **hibernate**. **Hibernate** uses this **file** to establish connection to the database server.It is an XML **file** which is used to define below information. Standard name for this **file** is **hibernate**. **cfg**. Xml

35.