JDBC

1. **What is JDBC?**
   JDBC is a layer of abstraction that allows users to choose between databases. It allows you to change to a different database engine and to write to a single API. JDBC allows you to write database applications in Java without having to concern yourself with the underlying details of a particular database.

2. **What are the common tasks or steps of JDBC?**
   1. Create an instance of a JDBC driver or load JDBC drivers through jdbc.drivers;
   2. Register a driver;
   3. Specify a database;
   4. Open a database connection;
   5. Submit a query;
   6. Receive results

3. **What Class.forName will do while loading drivers of JDBC?**
   It is used to create an instance of a driver and register it with the DriverManager. When you have loaded a driver, it is available for making a connection with a DBMS.

4. **Difference between PreparedStatement & Statement?**
   preparedstatement extends statement with added advantage of taking the arguments at run time.
   preparedStatement pstmt =con.prepareStatement("update FIRST_TABLE set job_code = ? where name = ? ");
   pstmt.setInt(1,2); pstmt.setString(2,"JOHN");
   When the PreparedStatement is executed, the DBMS can just run the PreparedStatement 's SQL statement without having to compile it first.

   Statement is used to execute static queries in the databases. It can't take the parameters at run time.
   stmt.executeQuery("select * from FIRST_TABLE");

5. **What is SQLException?**
   The SQLException class and its subtypes provide information about errors and warnings that occur while a data source is being accessed.
   The base class for exceptions that occur while running JDBC applications is SQLException. Every method of the JDBC API is declared as being able to throw SQLExceptions.
   SQLException is an extension of java.lang.Exception and provides additional information related to failures that happen in a database context. Specifically, the following information is available from an SQLException:

- Text description
- SQLState
- Error code
- A reference to any other exceptions that also occurred

6. **What is ResultSet object?**
   A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set.
   A default ResultSet object is not updatable and has a cursor that moves forward only. Thus, we can iterate through it only once and only from the first row to the last row. It is possible to produce ResultSet objects that are scrollable and/or updatable.

7. **What packages are used by JDBC?**
   There are 8 packages: java.sql.Driver, Connection,Statement, PreparedStatement, CallableStatement, ResultSet, ResultSetMetaData, DatabaseMetaData.

8. **What is the difference between executequery () and executeupdate ()?**
   executeQuery() - is for operation select of Sql by PreparedStatement or Statement.
   executeUpdata()- is for the operations such as insert, update or delete on SQL by PreparedStatement or Statement.

9. **What is TableModel?**
   The TableModel interface specifies the methods the JTable will use to interrogate a tabular data model.
   The JTable can be set up to display any data model which implements the TableModel interface with a couple of lines of code:

```
TableModel myData = new TableModel();
JTable table = new JTable(myData);
```

10. **What is DefaultMutableTreeNode & DefaultTreeModel?**

    A DefaultMutableTreeNode is a general-purpose node in a tree data structure. DefaultMutableTreeNode provides operations for examining and modifying a node's parent and children and also operations for examining the tree that the node is a part of. This class provides enumerations for efficiently traversing a tree or subtree in various orders or for following the path between two nodes. A DefaultMutableTreeNode may also hold a reference to a user object, the use of which is left to the user. Asking a DefaultMutableTreeNode for its string representation with toString() returns the string representation of its user object.

    DefaultTreeModel is a simple tree data model that uses TreeNodes.

XML

11. What is XML?

XML is the Extensible Markup Language. It improves the functionality of the Web by letting you identify your information in a more accurate, flexible, and adaptable way.
It is extensible because it is not a fixed format like HTML (which is a single, predefined markup language). Instead, XML is actually a metalanguage a language for describing other languages which lets you design our own markup languages for limitless different types of documents. XML can do this because it's written in SGML, the international standard metalanguage for text document markup (ISO 8879).

12. Why is XML such an important development?

It removes two constraints which were holding back Web developments:
1. dependence on a single, inflexible document type (HTML) which was being much abused for tasks it was never designed for;
2. the complexity of full SGML, whose syntax allows many powerful but hard-to-program options.
XML allows the flexible development of user-defined document types. It provides a robust, non-proprietary, persistent, and verifiable file format for the storage and transmission of text and data both on and off the Web; and it removes the more complex options of SGML, making it easier to program for.

13. Describe the differences between XML and HTML?

Differences Between XML and HTML

XML
- ► User definable tags
- ► Content driven
- ► End tags required for well formed documents
- ► Quotes required around attributes values
- ► Slash required in empty tags

HTML
- ► Defined set of tags designed for web display
- ► Format driven
- ► End tags not required
- ► Quotes not required
- ► Slash not required

14. What is an XML namespace?

An XML namespace is a collection of names that can be used as element or attribute names in an XML document.
The namespace qualifies element names uniquely on the Web in order to avoid conflicts between elements with the same name.

15. What is DTD?

A DTD (Document Type Definition) allows us to:

a. Define a specific set of tags with specific relationships to one another
b. Define default values for attributes
c. Define additional text and binary entities, along with their associated notations
Indicate the starting (root) element

16. What is XML Schema?

The schema defines the elements that can appear within the document and the attributes that can be associated with an element.

It also defines the structure of the document: which elements are child elements of others, the sequence in which the child elements
can appear, and the number of child elements.

17. What is document object model?

The Document Object Model (DOM) is an interface specification maintained by the W3C DOM Workgroup that defines an application independent mechanism to access, parse, or update XML data. In simple terms it is a hierarchical model that allows developers to manipulate XML documents easily Any developer that has worked extensively with XML should be able to discuss the concept and use of DOM objects freely.

18. What is a Parser?

Parser is a software program that recognizes the rules of XML
In well-formed XML

a. Checks document to see if it follows the well-formedness rules

in Valid XML

b. Checks an XML DTD to see if it follows the rules of XML, then

Checks an XML document to see if it follows the rules of XML, and also adheres to the structure against its DTD

19. What is Well Formed XML Document?

A "Well Formed" XML document has correct XML syntax.
The syntax rules are:

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested

XML attribute values must be quoted

RMI

20. What is Java RMI?

Remote Method Invocation (RMI) is the process of activating a method on a remotely running object. RMI offers location transparency in the sense that it gives the feel that a method is executed on a locally running object.

21. What are the layers of RMI Architecture?

The RMI is built on three layers.

a. Stub and Skeleton layer

This layer lies just beneath the view of the developer. This layer intercepts method calls made by the client to the interface reference variable and redirects these calls to a remote RMI service.

b. Remote Reference Layer.

This layer understands how to interpret and manage references made from clients to the remote service objects. The connection is a one-to-one (unicast) link.

c. Transport layer

This layer is based on TCP/IP connections between machines in a network. It provides basic connectivity, as well as some firewall penetration strategies.

22. What is the role java.rmi.Naming Class?

The Naming class provides methods for storing and obtaining references to remote objects in the remote object registry.

23. What is the use of UnicastRemoteObject in RMI?

The UnicastRemoteObject class provides support for point-to-point active object references using TCP streams. Objects that require remote behavior should extend UnicastRemoteObject.

24. What is the difference between using bind() and rebind() methods of Naming Class?

bind method(String name) binds the specified name to a remote object while rebind(String name) method rebinds the specified name to a new remote object,any existing binding for the name is replaced.

25. What is Stub & skeleton?

A stub is a proxy for a remote object that runs on the client computer.

A skeleton is a proxy for a remote object that runs on the server.

Stubs forward a client's remote method invocations (and their associated arguments) to skeletons, which forward them on to the appropriate server objects. Skeletons return the results of server method invocations to clients via stubs.