JEE L3 (1 hour) Th-Fri

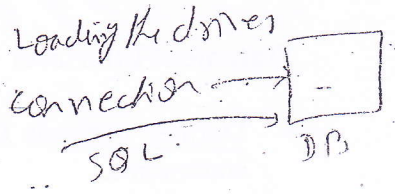
JDBC -1

JDBC

- The JDBC (Java Database Connectivity) API helps a **java Program to access a database in a standard way.**
- JDBC is a **specification** that tells the database vendors how to write a driver program to interface java programs with their database.
- A Driver written according to this standard is called the **JDBC Driver.**
- All related classes and interfaces are present in the **java.sql** package.
- All JDBC Drivers **implement the interfaces** of java.sql.

Database interaction:

- The steps involved in a database interaction are:
  - **Loading** the specific driver
  - Making a **connection** to the database
  - **Sending** SQL statements to the database
  - **Processing** the results



JDBC Drivers:

- There are four types of drivers:
  1. JDBC-ODBC Bridge Driver
  2. Native-API Driver
  3. Network-protocol Driver
  4. Native-Protocol Driver

L
C
S
P

Statement:

- A statement object is used to send SQL statements to a database. Three kind of statement objects are there:
  1. **Statement:** Execute simple SQL without parameters.
  2. **PreparedStatement:** Used for precompiled SQL statements with or without parameters.
  3. **CallableStatement:** Execute a call to a database stored procedure or function.

JDBC Classes and Interfaces:

DriverManager Class:

- Manages all the JDBC Drivers that are loaded in the memory
- Helps in dynamic loading of Drivers.

Connection Interface: Defines methods for interacting with the database via the established connection. → Set connections

- A connection object represents a connection with a database.
- A connection session includes the SQL statements that are executed and the results that are returned over that connection.
- A single application can have one or more connections with a single database, or it can have many connections with many different databases.

Statement Interface:

- Defines methods that are used to interact with database via the execution of SQL statements.

ResultSet Interface:

- Maintains a pointer to a row within the tabular results. The next() method is used to successfully step through the rows of the tabular results.

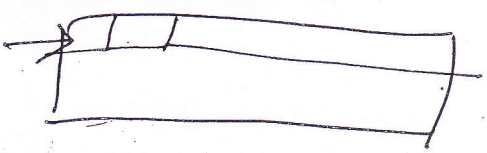
PreparedStatement interface: helps us to work with precompiled SQL statements.

- Precompiled SQL statements are faster than normal statements
- So if a SQL statement is to be repeated, it is better to use PreparedStatement

CallableStatement interface: helps us to call stored procedures and functions.

java.sql.packa

Connection
Statement
Result Set
Prepared Statement
Callable Statement



JDBC-3

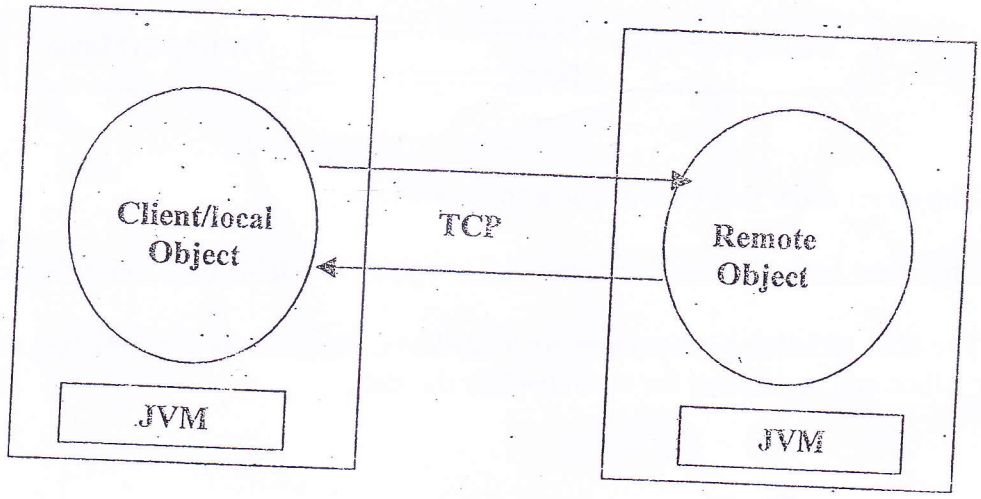
Remote Method Invocation

- Remote Method Invocation enables applications written in the Java programming language to interoperate with each other in a transparent, flexible way, even when deployed on very different systems.

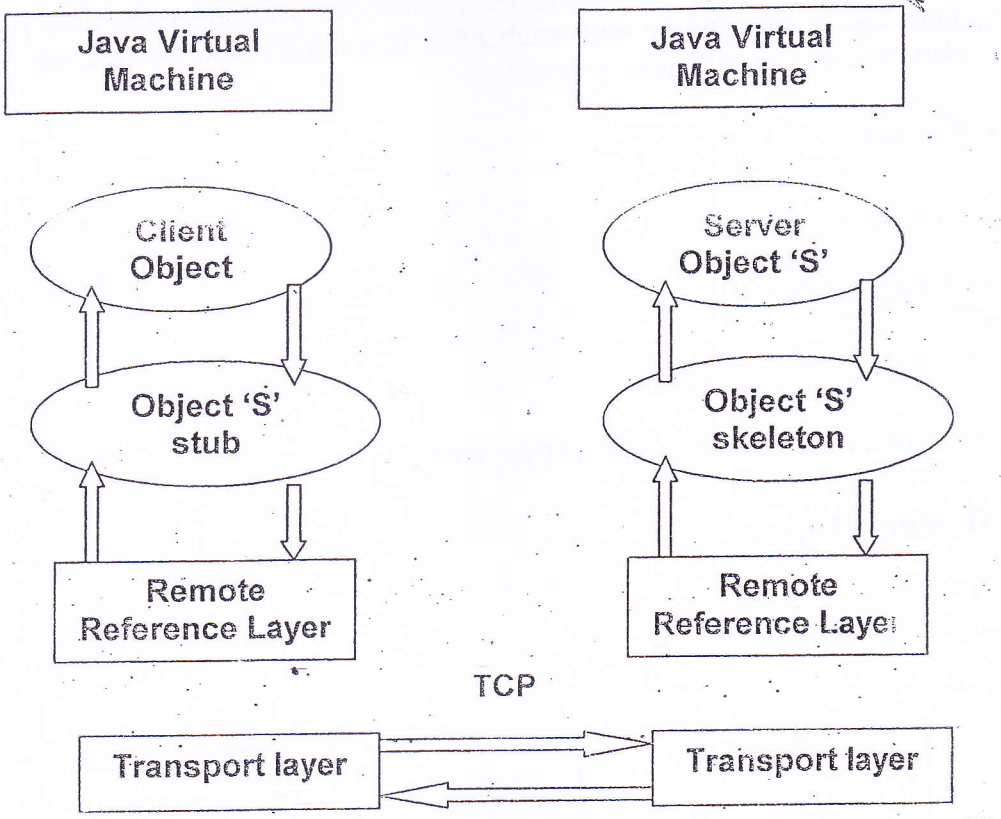
1. Example of Use:

- Database access
- Computations
- Any custom protocol
- Not for standard protocols (HTTP, FTP, etc)

2. Remote Objects:



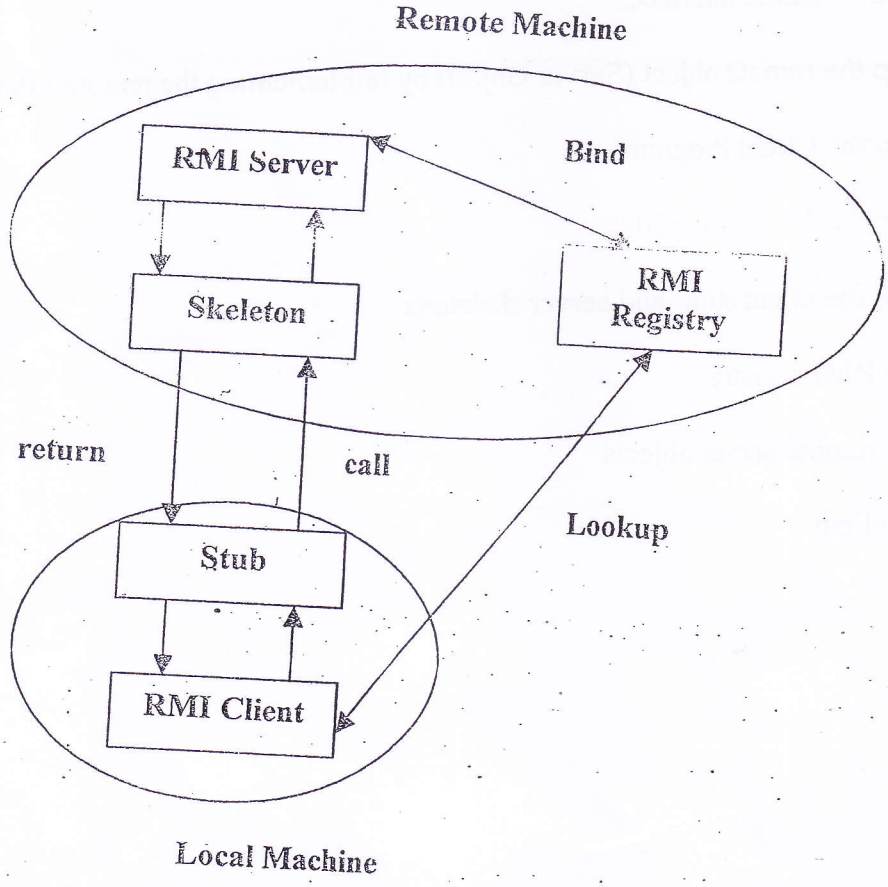
3. RMI Layers:



1. The server must first bind its name to the **registry**.
2. The client **lookup** the server name in the registry to establish remote references.
3. The Stub serializing the parameters to skeleton, the skeleton invoking the remote method and serializing the result back to the stub.

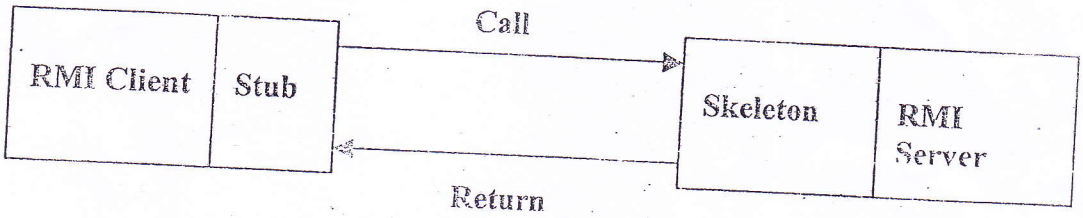
3DBC5

4. General RMI Architecture:



Stub and Skeleton:

1. A client invokes a remote-method, **the call is first forwarded to stub.**
2. The stub is responsible for sending the remote call over to the server-side skeleton.
3. The stub opening a socket to the remote server, marshalling the object parameters and forwarding the data stream to the skeleton.
4. A skeleton contains a method that receives the remote calls, unmarshals the parameters, and invokes the actual remote object implementation.



JDBC 6

5. Steps to Develop a RMI System:

- Define the remote interface.
- Develop the remote object (Server Object) by implementing the remote interface.
- Develop the Client Program.
- Compile the Java source files.
- Generate the client stubs and server skeletons.
- Start the RMI registry
- Start the remote server objects
- Run the client