Unit V & VI Notes - Network Programming and JDBC

Unit V: Basics of Network Programming

Unit V: Basics of Network Programming

5.1 Socket Overview

Client/Server Model:

- Client initiates communication.

- Server waits and responds.

Reserved Sockets:

- Port ranges: 0-1023 (well-known), 1024-49151 (registered), 49152-65535 (dynamic).

Proxy Servers:

- Acts as middleman, caching, access control, hides internal IPs.

Internet Addressing:

- IPv4 (32-bit), IPv6 (128-bit).

- DNS translates domain names to IPs.

5.2 Java and the Net

Java Networking Classes (java.net):

- InetAddress, Socket, ServerSocket, DatagramSocket, DatagramPacket, URL, URLConnection.

InetAddress Class:

Factory Methods:

- getByName(), getLocalHost(), getAllByName()

Instance Methods:

- getHostName(), getHostAddress(), toString()

5.3 Socket Programming

TCP/IP Sockets:

Client:

- Socket s = new Socket("host", port);

Server:

- ServerSocket ss = new ServerSocket(port); Socket s = ss.accept();

UDP Sockets:

Client:

- DatagramSocket, DatagramPacket for sending.

Server:

- DatagramSocket.receive()

Uniform Resource Locator

5.4 URL and URLConnection Classes

URL Class:

- Represents web address. Methods: getProtocol(), getHost(), getFile(), openStream()

URLConnection Class:

- Represents connection to resource. Use openConnection(), getInputStream()

Unit VI: Interacting with Database

Unit VI: Interacting with Database

6.1 JDBC and ODBC

ODBC:

- Microsoft-based, uses ODBC drivers, platform-independent.

JDBC:

- Java API for databases, cross-platform.

6.2 JDBC Architecture

Two-Tier:

- Client directly interacts with DB.

Three-Tier:

- Client -> Middleware -> DB. More scalable and secure.

6.3 JDBC Drivers

Type 1: JDBC-ODBC Bridge - Deprecated, platform-dependent.

Type 2: Native API - Uses native libraries.

Type 3: Network Protocol - Via middleware server.

Type 4: Thin Driver - Pure Java, direct DB protocol.

JDBC Key Classes and Interfaces:

Class Class:

- Loads JDBC driver: Class.forName("...")

DriverManager:

- getConnection(url, user, pass)

Connection Interface:

- Represents active DB connection.

Statement Interface:

- Executes static SQL.

PreparedStatement Interface:

- Executes parameterized queries, prevents SQL injection.

ResultSet Interface:

- Iterates over results: next(), getInt(), getString(), etc.