Name- Sachin Kumar Singh, Roll No.- 41221138, Sub.- Java Programming

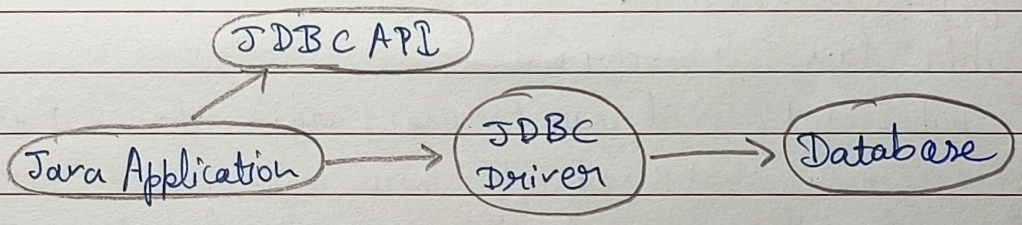
Topic ASSIGNMENT - 3 Date

Q1 Discuss JDBC Architecture. Explain the steps to connect JAVA application to Database.

Ans The JDBC API supports both two-ties & three-ties procesing models for database accers but in general JDBC architecture consists of two layers -

i) JDBC API :- This provides the application - to - JDBC manager connection.

ii) JDBC Driver API :- This supports the JDBC Manager - to - Driver connection



There are 5 steps to connect any java application with the database using JDBC :-

i) Register the Driver class - The forName() method of 'class' class is used to register the driver class.
Syntax - public static void forName(String className) throws ClassNotFoundException

ii) Create the connection object - The getConnection() method of DriverManager class is used to establish connection with database.

iii) Create the statement object - This method of connection interface is used to create statement.
Syntax - public Statement createStatement() throws SQLException.

iv) Execute the Query - This method is used to execute queries to the database.
Syntax - public ResultSet executeQuery(String sq) throws SQLException.

Topic _____ Date _____

v) Close the connection object - By closing connection object statement will be closed automatically.
Syntax - public void close() throws SQL Exception.

Q2 WAP to make a chat program by client side and server side socket programming.

Ans Server side coding for Message Passing (Two-way communication):

```
import.java.io.*;
import.java.net.*;
public class TestServer {
    public static void main (String[] args) throws Exception
    { ServerSocket.sersock = new ServerSocket(3000);
      System.out.println("Server ready for chatting!");
      Socket sock = sersock.accept();
      BufferedReader keyRead = new BufferedReader (
                  new InputStreamReader (System.in));
      OutputStream ostream = sock.getoutput Stream();
      PrintWriter pwrite = new PrintWriter (ostream, true);
      InputStream istream = sock.getInput Stream();
      BufferedReader receiveRead = new BufferedReader (
                  new InputStreamReader (istream));
      String receiveMessage, sendMessage;
      while (true){
        if ((receiveMessage = receiveRead.readline()!= NULL) {
            System.out.println (receivedMessage);
        }
        SendMessage = keyRead.readline();
        pwrite.println (sendMessage);
      }
    }
}
```

Topic _____ Date _____

Client Side Coding for Message Passing :

```
import java.net.*;
import java.io.*;
public class TestClient {
    public static void main(String[] args) {
        Socket sock = new Socket("127.0.0.1", 3000);
        BufferedReader keyRead = new BufferedReader(new
                BufferedReader(new InputStreamReader(System.In)));
        OutputStream OStream = sock.getOutputStream();
        PrintWriter pwrite = new printwrite(OStream, true);
        InputStream istream = sock.getInputStream();
        BufferedReader receivedRead = new BufferedReader(new
                InputStreamReader(istream));
        System.out.println("Start the Chitchat!!");
        String receivedMessage, sendMessage;
        While(true) {
            SendMessage = keyRead.readline();
            pwrite.println(SendMessage);
            pwrite.flush();
            if((receiveMessage = receiveRead.readline()) != NULL) {
                System.out.println(receive.Message);
            }
        }
    }
}
```

Q2 Write a simple program for JAVA database connectivity & query run.

Ans

```
import java.sql.*;
public class MySqlCon {
    public static void main(String [] args) {
        try {
            class.forName("com.mysql.jdbc.Driver");
```

Topic _____ Date _____

```
con = DriverManager.getConnection ("jdbc:mysql://
        localhost:3306/mydb", "root", "");
Statement stmt = con.createStatement();
while (rs.next()) {
    System.out.println(rs.getInt(1)+" "+rs.getString(2)
        +" "+rs.getString(3));
  } con.close();
} catch (Exception e) {
    System.out.println(e); }
  }
}
```

Q4 WAP to implement client side & server-side socket coding for creating connection & maintain communication between client & server.

Ans
```
import java.net.*;
import java.io.*;
public class Client {
    public static void main(String[] args) throws IOException {
        System.out.println("Client Started");
        Socket socket = new Socket("localhost", 1234); }
    }
import java.net.*;
import java.net.*;
public class Server {
    public static void main(String[] args) throws IOException {
        System.out.println("waiting for the client....");
        ServerSocket serverSocket = new ServerSocket(1234);
        Socket socket = ServerSocket.accept(); //Established connection
        System.out.println("Connection Established");
    }
}
```