**Exp no. 5 HALF DUPLEX CHAT USING TCP/IP**

**AIM:** To implement a chat server and client in java using TCP sockets in half duplex mode.

**DESCRIPTION:**

TCP Clients send requests to the server and the server will receive the request and response with acknowledgement. Every time either a client or a server can send and receive the messages

**ALGORITHM:**

**Server**

1. Create a server socket and bind it to the port.
2. Listen for new connections and when a connection arrives, accept it.
3. Read Client's message and display it
4. Get a message from user and send it to client
5. Repeat steps 3-4 until the client terminates
6. Close all streams
7. Close the server and client socket
8. Stop

**Client**

1. Create a client socket and connect it to the server’s port number

2. Get a message from user and send it to server

3. Read server's response and display it

4. Repeat steps 2-3 until chat is terminated with "exit" message

5. Close all input/output streams

6. Close the client socket

7. Stop

**Server**

import java.io.\*;

import java.net.\*;

class Server\_HalfDup {

    public static void main(String args[])

        throws Exception

    {

        // Create server Socket

        ServerSocket ss = new ServerSocket(888);

        // connect it to client socket

        Socket s = ss.accept();

        System.out.println("Connection established");

        // to send data to the client

        PrintStream ps

            = new PrintStream(s.getOutputStream());

        // to read data coming from the client

        BufferedReader br

            = new BufferedReader(

                new InputStreamReader(

                    s.getInputStream()));

        // to read data from the keyboard

        BufferedReader kb

            = new BufferedReader(

                new InputStreamReader(System.in));

        // server executes continuously

        while (true) {

            String str, str1;

            // repeat as long as the client

            // does not send a null string

            // read from client

            while ((str = br.readLine()) != null) {

                System.out.println("From Client:"+str);

                str1 = kb.readLine();

                // send to client

                ps.println(str1);

            }

            // close connection

            ps.close();

            br.close();

            kb.close();

            ss.close();

            s.close();

            // terminate application

            System.exit(0);

        } // end of while

    }

}

**Client**

import java.io.\*;

import java.net.\*;

class Client\_HalfDup {

    public static void main(String args[])

        throws Exception

    {

        // Create client socket

        Socket s = new Socket("localhost", 888);

        // to send data to the server

        DataOutputStream dos = new DataOutputStream(s.getOutputStream());

        // to read data coming from the server

        BufferedReader br

            = new BufferedReader(

                new InputStreamReader(

                    s.getInputStream()));

        // to read data from the keyboard

        BufferedReader kb

            = new BufferedReader(

                new InputStreamReader(System.in));

        String str, str1;

        // repeat as long as exit

        // is not typed at client

        while (!(str = kb.readLine()).equals("exit")) {

            // send to the server

            dos.writeBytes(str + "\n");

            // receive from the server

            str1 = br.readLine();

            System.out.println("From Server: "+str1);

        }

        // close connection.

        dos.close();

        br.close();

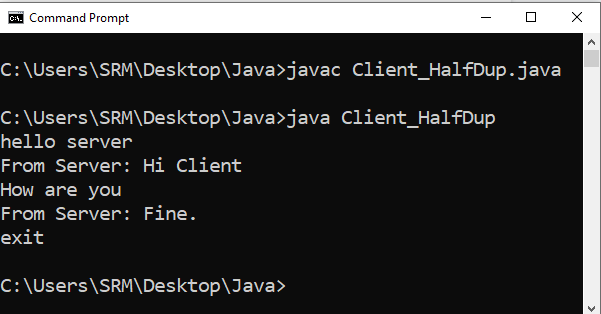
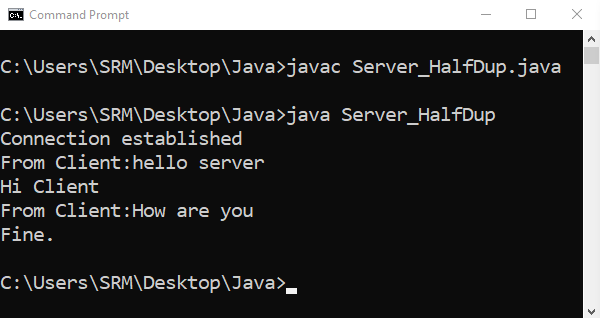
        kb.close();

        s.close();

    }

}

**Output**



**Result**: Thus Half Duplex Chat Using TCP/IP has been executed using Java programming