**CNT Assignment 1**

**Name: Atharva Salitri**

**Roll No.: 029**

**Branch: TY CSAI-B**

**Batch: B2**

**PRN: 12310120**

**Title:** **Write the client server programs using TCP Berkeley socket primitives for wired /wireless**

**network for following**

**a. to say Hello to Each other**

import java.io.\*;

import java.net.\*;

public class Client {

    public static void main(String[] args) throws Exception {

        System.out.println("Connecting to server...");

        Socket socket = new Socket("localhost", 5000);

        System.out.println("Connected to server.");

        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

        BufferedReader user = new BufferedReader(new InputStreamReader(System.in));

        String line, resp;

        while (true) {

            System.out.print("Client says: ");

            line = user.readLine();

            out.println(line);

            if (line.equals("exit")) break;

            resp = in.readLine();

            if (resp == null || resp.equals("exit")) break;

            System.out.println("Server says: " + resp);

        }

        socket.close();

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

    }

}

import java.io.\*;

import java.net.\*;

public class Server {

    public static void main(String[] args) throws Exception {

        ServerSocket server = new ServerSocket(5000);

        System.out.println("Server started, waiting for connection...");

        Socket socket = server.accept();

        System.out.println("Client connected.");

        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

        BufferedReader user = new BufferedReader(new InputStreamReader(System.in));

        String line, resp;

        while (true) {

            line = in.readLine();

            if (line == null || line.equals("exit")) break;

            System.out.println("Client says: " + line);

            System.out.print("Server says: ");

            resp = user.readLine();

            out.println(resp);

            if (resp.equals("exit")) break;

        }

        socket.close();

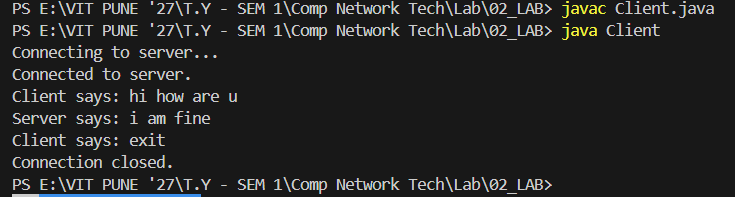
        server.close();

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

    }

}

OUTPUT:



**b. File transfer**

import java.io.\*;

import java.net.\*;

public class Client {

    public static void main(String[] args) throws Exception {

        System.out.println("Connecting to server...");

        Socket socket = new Socket("localhost", 9000);

        System.out.println("Connected to server.");

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

        File file = new File("E:\\VIT PUNE '27\\T.Y - SEM 1\\Comp Network Tech\\Lab\\02\_LAB\\FileTransfer\\client\_folder\\t.txt");

        FileInputStream fis = new FileInputStream(file);

        System.out.println("Sending file: " + file.getName() + " (" + file.length() + " bytes)");

        dos.writeLong(file.length());

        byte[] buffer = new byte[4096];

        int read;

        long totalSent = 0;

        while ((read = fis.read(buffer)) > 0) {

            dos.write(buffer, 0, read);

            totalSent += read;

            System.out.println("Sent " + totalSent + " of " + file.length() + " bytes.");

        }

        System.out.println("File sent successfully.");

        fis.close();

        dos.close();

        socket.close();

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

    }

}

import java.io.\*;

import java.net.\*;

public class Server {

    public static void main(String[] args) throws Exception {

        System.out.println("Waiting for connection...");

        ServerSocket serverSocket = new ServerSocket(9000);

        Socket socket = serverSocket.accept();

        System.out.println("Client connected.");

        DataInputStream dis = new DataInputStream(socket.getInputStream());

        FileOutputStream fos = new FileOutputStream(

            "E:\\VIT PUNE '27\\T.Y - SEM 1\\Comp Network Tech\\Lab\\02\_LAB\\FileTransfer\\server\_folder\\t.txt"

        );

        long fileSize = dis.readLong();

        System.out.println("Receiving file of size: " + fileSize + " bytes.");

        byte[] buffer = new byte[4096];

        int read;

        long totalRead = 0;

        while ((read = dis.read(buffer, 0, Math.min(buffer.length, (int)(fileSize - totalRead)))) > 0) {

            fos.write(buffer, 0, read);

            totalRead += read;

            System.out.println("Received " + totalRead + " of " + fileSize + " bytes.");

            if (totalRead == fileSize) break;

        }

        System.out.println("File received and saved to server\_folder/t.txt");

        fos.close();

        dis.close();

        socket.close();

        serverSocket.close();

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

    }

}

Output:

