|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap  Id: | 60004220127 |
| Class: | S. Y. B.Tech (Computer Engineering) |
| Course: | Computer Networks (DJ12CEL405) |
| Date of Performance: |  |
| Date of Submission: |  |
| Experiment  No.: | 06 |
| Aim: | Socket TCP and UDP |

**AIM: TO IMPLEMENT TCP AND UDP SOCKET COMMUNICATION IN JAVA.**

**TCP:**

**CODE:**

**Server.java**

// A Java program for a Server

import java.net.\*;

import java.io.\*;

public class Server {

    // initialize socket and input stream

    private Socket socket = null;

    private ServerSocket server = null;

    private DataInputStream in = null;

    private DataInputStream input = null;

    private DataOutputStream out = null;

    // constructor with port

    public Server(int port) {

        // starts server and waits for a connection

        try {

            server = new ServerSocket(port);

            System.out.println("Server started");

            System.out.println("Waiting for a client ...");

            socket = server.accept();

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

            // takes input from the client socket

            in = new DataInputStream(

                    new BufferedInputStream(socket.getInputStream()));

            out = new DataOutputStream(

                    socket.getOutputStream());

            input = new DataInputStream(System.in);

            String line = "";

            // reads message from client until "Over" is sent

            while (!line.equals("Over")) {

                try {

                    line = in.readUTF();

                    System.out.println(line);

                    line = input.readLine();

                    out.writeUTF(line);

                } catch (IOException i) {

                    System.out.println(i);

                }

            }

            System.out.println("Closing connection");

            // close connection

            socket.close();

            in.close();

        } catch (IOException i) {

            System.out.println(i);

        }

    }

    public static void main(String args[]) {

        Server server = new Server(5000);

    }

}

**Client.java**

// A Java program for a Client

import java.io.\*;

import java.net.\*;

public class Client {

    // initialize socket and input output streams

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream out = null;

    private DataInputStream in = null;

    // constructor to put ip address and port

    public Client(String address, int port) {

        // establish a connection

        try {

            socket = new Socket(address, port);

            System.out.println("Connected");

            // takes input from terminal

            input = new DataInputStream(System.in);

            // sends output to the socket

            out = new DataOutputStream(

                    socket.getOutputStream());

            in = new DataInputStream(

                    new BufferedInputStream(socket.getInputStream()));

        } catch (UnknownHostException u) {

            System.out.println(u);

            return;

        } catch (IOException i) {

            System.out.println(i);

            return;

        }

        // string to read message from input

        String line = "";

        // keep reading until "Over" is input

        while (!line.equals("Over")) {

            try {

                line = input.readLine();

                out.writeUTF(line);

                line = in.readUTF();

                System.out.println(line);

            } catch (IOException i) {

                System.out.println(i);

            }

        }

        // close the connection

        try {

            input.close();

            out.close();

            socket.close();

        } catch (IOException i) {

            System.out.println(i);

        }

    }

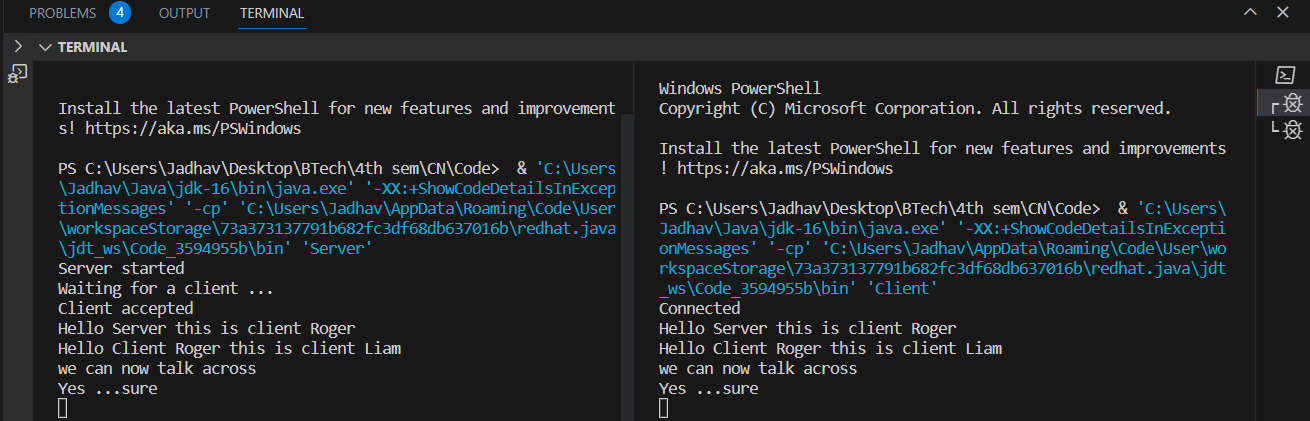
    public static void main(String args[]) {

        Client client = new Client("127.0.0.1", 5000);

    }

}

**OUTPUT:**

****

**UDP:**

**CODE:**

**Server.java:**

import java.io.\*;

import java.net.\*;

class Server {

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

        DatagramSocket serverSocket = new DatagramSocket(9876);

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

        byte[] receiveData = new byte[1024];

        byte[] sendData = new byte[1024];

        while (true) {

            DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

            serverSocket.receive(receivePacket);

            String sentence = new String(receivePacket.getData());

            System.out.println("RECEIVED: " + sentence);

            InetAddress IPAddress = receivePacket.getAddress();

            int port = receivePacket.getPort();

            String sentSentence = inFromServer.readLine();

            // String capitalizedSentence = sentence.toUpperCase();

            sendData = sentSentence.getBytes();

            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, port);

            serverSocket.send(sendPacket);

            receiveData = new byte[1024];

            if (sentSentence.equalsIgnoreCase("bye")) {

                serverSocket.close();

                break;

            }

        }

    }

}

**Client.java:**

import java.io.\*;

import java.net.\*;

class Client {

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

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

        DatagramSocket clientSocket = new DatagramSocket();

        InetAddress IPAddress = InetAddress.getByName("localhost");

        byte[] sendData = new byte[1024];

        byte[] receiveData = new byte[1024];

        while (true) {

            String sentence = inFromUser.readLine();

            sendData = sentence.getBytes();

            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

            clientSocket.send(sendPacket);

            DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

            clientSocket.receive(receivePacket);

            String modifiedSentence = new String(receivePacket.getData());

            System.out.println("FROM SERVER:" + modifiedSentence);

            receiveData = new byte[1024];

            if (sentence.equalsIgnoreCase("bye")) {

                clientSocket.close();

                break;

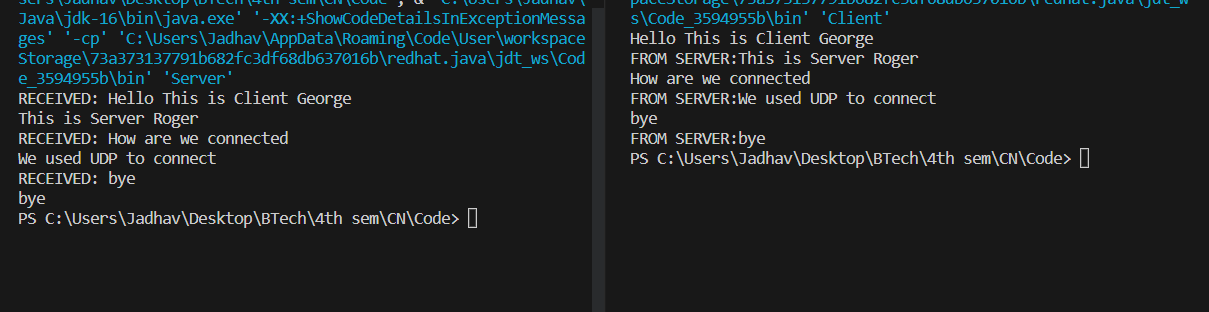
            }

        }

    }

}

**OUTPUT:**

****