**Lab Practical #07:**

Study Client-Server Socket programming - TCP & UDP

**Practical Assignment #07:**

1. **Write a C/Java code for TCP Server-Client Socket Programming.**
2. **Write a C/Java code for UDP Server-Client Socket Programming.**
3. **For TCP Server-Client:**

**TCP Server Program:**

import java.io.\*;

import java.net.\*;

public class TCPServer {

    public static void main(String[] args) {

        try {

            // Create a server socket that listens on port 5000

            ServerSocket serverSocket = new ServerSocket(5000);

            System.out.println("Server is listening on port 5000...");

            // Wait for a client to connect

            Socket socket = serverSocket.accept();

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

            // Create input and output streams to communicate with the client

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

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

            // Loop to continuously exchange messages

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

            String receivedMessage = "";

            String sendMessage = "";

            while (!receivedMessage.equals("bye")) {

                // Receive message from client

                receivedMessage = input.readUTF();

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

                // Send a response to the client

                sendMessage = reader.readLine();

                output.writeUTF(sendMessage);

                output.flush();

            }

            // Close resources

            input.close();

            output.close();

            socket.close();

            serverSocket.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

**TCP Client Program:**

import java.io.\*;

import java.net.\*;

public class TCPClient {

    public static void main(String[] args) {

        try {

            // Connect to the server running on localhost at port 5000

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

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

            // Create input and output streams to communicate with the server

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

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

            // Loop to continuously exchange messages

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

            String sendMessage = "";

            String receivedMessage = "";

            while (!sendMessage.equals("bye")) {

                // Send a message to the server

                sendMessage = reader.readLine();

                output.writeUTF(sendMessage);

                output.flush();

                // Receive response from the server

                receivedMessage = input.readUTF();

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

            }

            // Close resources

            input.close();

            output.close();

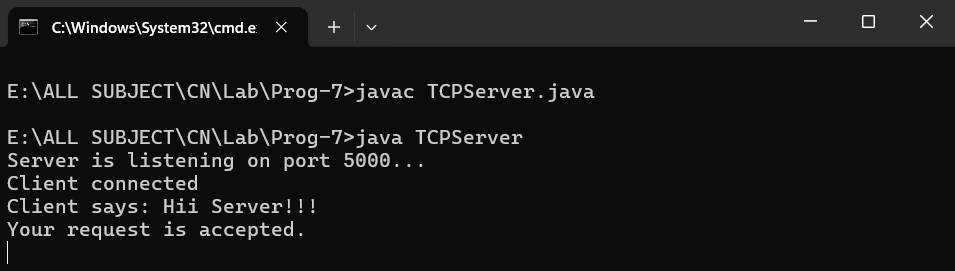
            socket.close();

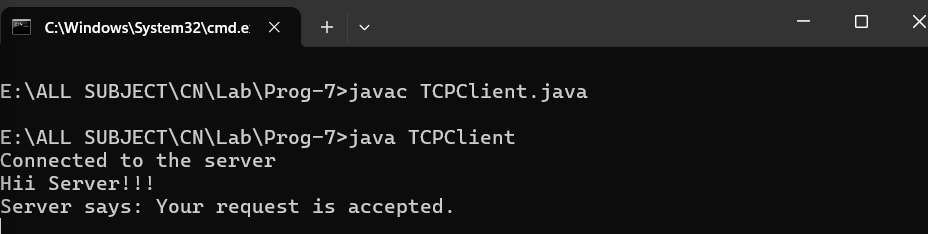
        } catch (IOException e) {

            e.printStackTrace();

        }

    }

****}

****

1. **For UDP Server-Client:**

**UDP Server Program:**

import java.net.\*;

public class UDPServer {

    public static void main(String[] args) {

        try {

            // Create a DatagramSocket to listen on port 5000

            DatagramSocket serverSocket = new DatagramSocket(5000);

            byte[] receiveBuffer = new byte[1024];

            byte[] sendBuffer;

            System.out.println("Server is listening on port 5000...");

            while (true) {

                // Create a DatagramPacket to receive data from the client

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

                serverSocket.receive(receivePacket);

                // Convert received data to a string and display it

                String receivedMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

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

                // Break the loop if the client sends "bye"

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

                    break;

                }

                // Send a response to the client

                InetAddress clientAddress = receivePacket.getAddress();

                int clientPort = receivePacket.getPort();

                String sendMessage = "Message received: " + receivedMessage;

                sendBuffer = sendMessage.getBytes();

        DatagramPacket sendPacket = new DatagramPacket(sendBuffer, sendBuffer.length, clientAddress, clientPort);

                serverSocket.send(sendPacket);

            }

            // Close the socket

            serverSocket.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

**UDP Client Program:**

import java.net.\*;

import java.io.\*;

public class UDPClient {

    public static void main(String[] args) {

        try {

            // Create a DatagramSocket

            DatagramSocket clientSocket = new DatagramSocket();

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

            byte[] sendBuffer;

            byte[] receiveBuffer = new byte[1024];

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

            while (true) {

                // Read a message from the user

                System.out.print("Enter message: ");

                String sendMessage = reader.readLine();

                sendBuffer = sendMessage.getBytes();

                // Create a DatagramPacket to send the data to the server

                DatagramPacket sendPacket = new DatagramPacket(sendBuffer, sendBuffer.length, serverAddress, 5000);

                clientSocket.send(sendPacket);

                // Break the loop if the message is "bye"

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

                    break;

                }

                // Receive a response from the server

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

                clientSocket.receive(receivePacket);

                String receivedMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

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

            }

            // Close the socket

            clientSocket.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

