**NAME: SNEHA TIWARI**

**ROLL NO.: 11500119052**

## ASSIGNMENT 4: Connection oriented UDP socket programming (Two Way Communication) => Chatting + Problem Solving

## Client Side: udpClient.java

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.util.Scanner;

public class udpClient {

    public static void main(String args[]) throws IOException

    {

        Scanner sc = new Scanner(System.in);

        DatagramSocket ds = new DatagramSocket();

        InetAddress ip = InetAddress.getLocalHost();

        byte buf[] = null;

        byte buf2[] = null;

        System.out.println("Enter text: ");

        while (true)

        {

            String inp = sc.nextLine();

            buf2 = inp.getBytes();

            DatagramPacket DpSend = new DatagramPacket(buf2, buf2.length, ip, 1234);

            ds.send(DpSend);

            if (inp.equals("Goodbye"))

                break;

            System.out.print("Enter the equation in the format:");

            System.out.println("'operand1 operator operand2'");

            inp = sc.nextLine();

            buf = new byte[65535];

            buf = inp.getBytes();

            DatagramPacket DpSend2 = new DatagramPacket(buf, buf.length, ip, 5678);

            ds.send(DpSend2);

            buf = new byte[65535];

            DatagramPacket DpReceive = new DatagramPacket(buf, buf.length);

            ds.receive(DpReceive);

            System.out.println("Answer: " + new String(buf, 0, buf.length));

        }

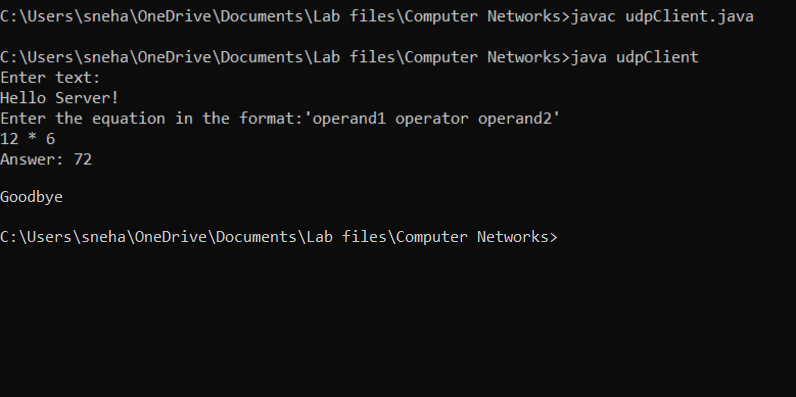
        sc.close();

        ds.close();

    }

}

**Output:**

****

**Server Side: udpServer.java**

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.lang.StringBuilder;

import java.util.StringTokenizer;

class udpServer {

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

        DatagramSocket ds = new DatagramSocket(1234);

        byte[] buf = null;

        DatagramPacket DpReceive = null;

        DatagramPacket DpSend = null;

        DatagramSocket ds2 = new DatagramSocket(5678);

        byte[] receive = new byte[65535];

        DatagramPacket DpReceive2 = null;

        while (true) {

            DpReceive = new DatagramPacket(receive, receive.length);

            ds.receive(DpReceive);

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

            if (data(receive).toString().equals("Goodbye"))

            {

                System.out.println("\nClient sent Goodbye.\nExiting....");

                break;

            }

            receive = new byte[65535];

            buf = new byte[65535];

            DpReceive2 = new DatagramPacket(buf, buf.length);

            ds2.receive(DpReceive2);

            String inp = new String(buf, 0, buf.length);

            inp = inp.trim();

            System.out.println("Equation Received: " + inp);

            int result;

            StringTokenizer st = new StringTokenizer(inp);

            if (inp.equals("Goodbye")) {

                System.out.println("\nClient sent Goodbye.\nExiting....");

                break;

            }

            int oprnd1 = Integer.parseInt(st.nextToken());

            String operation = st.nextToken();

            int oprnd2 = Integer.parseInt(st.nextToken());

            if (operation.equals("+"))

                result = oprnd1 + oprnd2;

            else if (operation.equals("-"))

                result = oprnd1 - oprnd2;

            else if (operation.equals("\*"))

                result = oprnd1 \* oprnd2;

            else

                result = oprnd1 / oprnd2;

            System.out.println("Sending the result...");

            String res = Integer.toString(result);

            buf = res.getBytes();

            int port = DpReceive.getPort();

            DpSend = new DatagramPacket(buf, buf.length, InetAddress.getLocalHost(), port);

            ds2.send(DpSend);

        }

        ds2.close();

        ds.close();

    }

    public static StringBuilder data(byte[] a) {

        if (a == null)

            return null;

        StringBuilder ret = new StringBuilder();

        int i = 0;

        while (a[i] != 0) {

            ret.append((char) a[i]);

            i++;

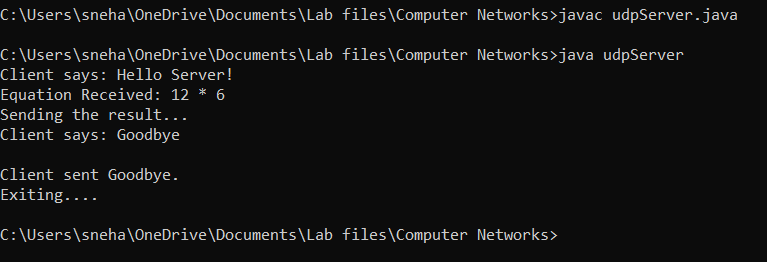
        }

        return ret;

    }

}

**Output:**

****

**----------------------------------------------------------------**