**Name:** Mitrajeet Golsangi

**Roll No:** 01

**PRN:** 12010484

**Division:** Ty-CS-B

**Batch:** B1

**TY. B. Tech.**

**CS3052: Computer Networks**

**Lab No: 6**

**Develop a client server using UDP Berkeley socket primitives for chat applications in peer-to-peer and client-server mode. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer-to-peer mode.**

**Code: (Client)**

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.InetAddress;

import java.net.Socket;

import java.net.UnknownHostException;

public class client\_udp {

public static void main(String[] args) {

      try {

        InetAddress host = InetAddress.getLocalHost();

    Socket socket = new Socket(host.getHostName(), 6000);

    DataOutputStream dos = new DataOutputStream(

            socket.getOutputStream());

        DataInputStream dis = new DataInputStream(System.in);

    String line = "";

    while (!line.equals("bye"))

    {

       try

       {

        line = dis.readLine();

            dos.writeUTF(line);

        dos.flush();

       }

       catch(IOException ioe)

       {

        System.out.println("Sending error: " + ioe.getMessage());

                }

       }

           dis.close();

       dos.close();

    } catch (UnknownHostException e) {

        e.printStackTrace();

    } catch (IOException e) {

        e.printStackTrace();

    }

   }

}

**Code: (Server)**

import java.io.BufferedInputStream;

import java.io.DataInputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

public class server\_udp {

   private ServerSocket server;

   private int port = 6000;

   public server\_udp() {

      try {

    server = new ServerSocket(port);

      } catch (IOException e) {

       e.printStackTrace();

    }

   }

   public static void main(String[] args) {

    server\_udp server = new server\_udp();

    server.connection();

   }

   public void connection() {

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

    try

    {

           Socket socket = server.accept();

       System.out.println("Client accepted: " + socket);

       DataInputStream dis = new DataInputStream(

        new BufferedInputStream(socket.getInputStream()));

       boolean done = false;

       while (!done)

       {

        try

        {

               String line = dis.readUTF();

           System.out.println(line);

           done = line.equals("bye");

        }

        catch(IOException ioe)

        {

           done = true;

        }

           }

        dis.close();

        socket.close();

       }

       catch(IOException ioe)

       {

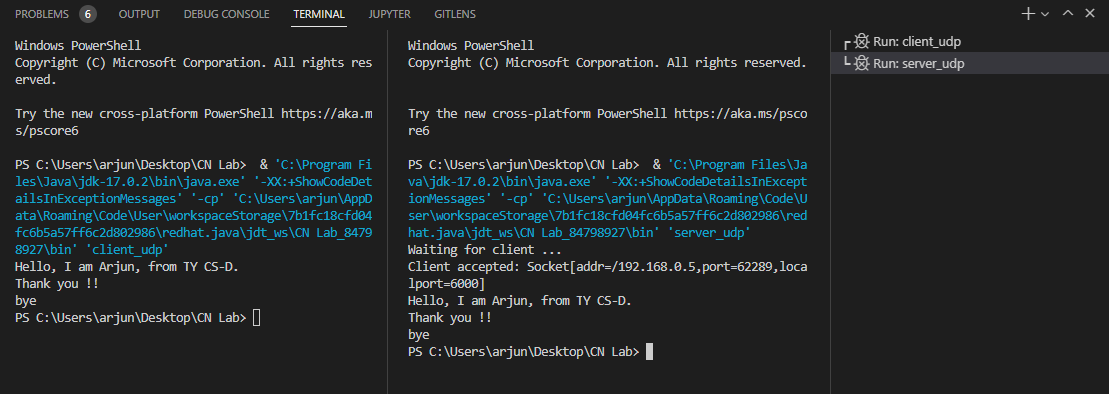
        System.out.println(ioe);

       }

    }

}

**Output:**

****