Thiagarajar College of Engineering, Madurai – 625015

Department of Information Technology

LAB RECORD

Lab Course Code	18IT470
Lab Course Name	Computer Networks Lab
Programme	B.Tech. (IT)
Semester	IV/Even
Academic Year	2020 – 21

Submitted By,

Student Name	:	**SREE VIGNESH K**
Register Number	:	**19IT102**

TABLE OF CONTENTS

EXP. NO. 1 **Simple Client-Server Chat Application Development**

AIM:

To create a server – client point to point communication application using java

PROCEDURE:

Server:

Step1:Create a server socket and bind it to port.

Step 2:Listen for new connection and when a connection arrives, accept it.

Step 3:Send server's date and time to the client.

Step 4:Read client's IP address sent by the client.

Step 5:Display the client details.

Step 6:Repeat steps 2-5 until the server is terminated.

Step 7:Close the server socket.

Client

Step1:Create a client socket and connect it to the server's port number.

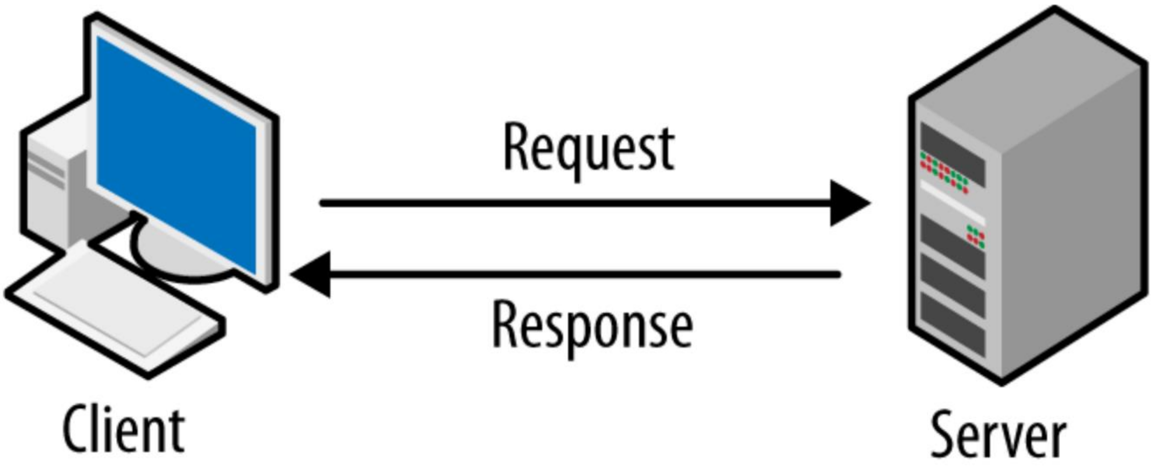
Step2:Retrieve its own IP address using built-in function.

Step3:Send its address to the server.

Step4:Display the date & time sent by the server.

Step5:Close the client socket

DIAGRAM:



CODE / IMPLEMENTATION:

TCP method:

Server:

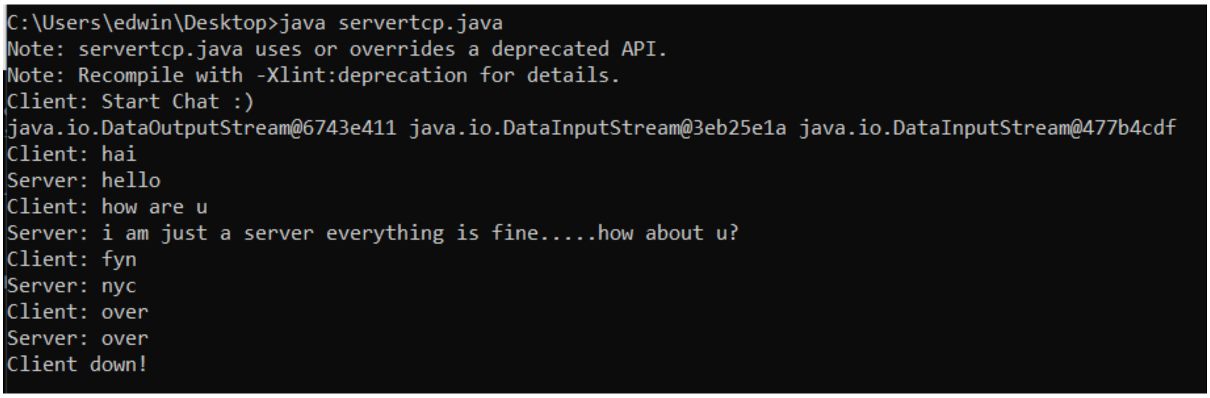
```
import java.io.*;

import java.net.*;

import java.util.Scanner;


public class Server {

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

        {

            ServerSocket ss=new ServerSocket(7888);

            Socket s = ss.accept();

            DataInputStream din=new
DataInputStream(s.getInputStream());

            String str;

            str=din.readUTF();

            System.out.println("Client:\t"+str);

            DataOutputStream dout=new
DataOutputStream(s.getOutputStream());

            DataInputStream msg=new DataInputStream(System.in);

            System.out.println(dout + " " + din + " "+ msg);

            while(true)

            {

                str=din.readUTF();

                System.out.print("Client:\t"+str);

                System.out.print("Server:\t");

                str=msg.readLine();

                dout.writeUTF(str);

                if(str.equals("over"))

                {

                    System.out.println("Client down!");
                try{
                    s.close();
                    break;
                }
                catch (IOException e){
                    System.out.println("Socket Closed :(");
                }
                }
                }
        }
}
```

Client:

```
        import java.io.*;
        import java.net.Socket;
        import java.util.Scanner;

        public class Client
        {
            public static void main(String[] args) throws Exception
            {
                Socket s=new Socket("127.0.0.1",7888);
                if(s.isConnected())
                {
                    System.out.println("Connected to server");
                }
                DataInputStream msg=new DataInputStream(System.in);
                String str="Start Chat :)";
                DataOutputStream dout=new
        DataOutputStream(s.getOutputStream());
                dout.writeUTF(str);
                System.out.println(str);

                DataInputStream din=new
        DataInputStream(s.getInputStream());

                System.out.println(dout + " " + din + " "+ msg);

                while(true)

                    {

                        System.out.print("Client:\t");

                        str=msg.readLine();

                        dout.writeUTF(str+"\n");

                        str=din.readUTF();

                        System.out.println("Server:\t"+str);

                        if(str.equals("over"))

                        {

                            System.out.println("Client down!");

                            try{

                                s.close();

                                break;

                            }

                            catch(IOException e){

                                System.out.println("Scoket Closed :(");

                            }

                        }


                    }

            }

        }
```

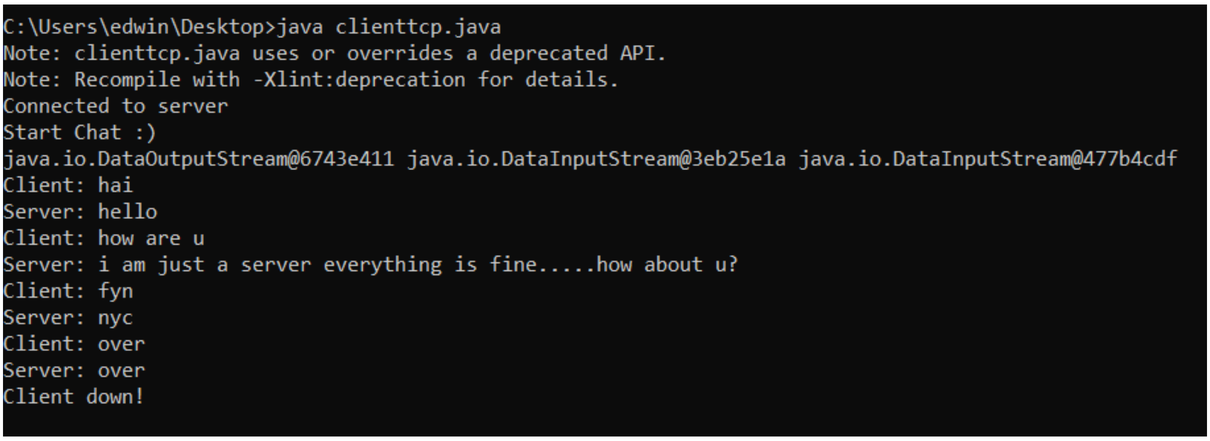
OUTPUT SCREENSHOTS:

Server:



```
C:\Users\edwin\Desktop>java servertcp.java
Note: servertcp.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Client: Start Chat :)
java.io.DataOutputStream@6743e411 java.io.DataInputStream@3eb25e1a java.io.DataInputStream@477b4cdf
Client: hai
Server: hello
Client: how are u
Server: i am just a server everything is fine.....how about u?
Client: fyn
Server: nyc
Client: over
Server: over
Client down!
```

Client:



```
C:\Users\edwin\Desktop>java clienttcp.java
Note: clienttcp.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected to server
Start Chat :)
java.io.DataOutputStream@6743e411 java.io.DataInputStream@3eb25e1a java.io.DataInputStream@477b4cdf
Client: hai
Server: hello
Client: how are u
Server: i am just a server everything is fine.....how about u?
Client: fyn
Server: nyc
Client: over
Server: over
Client down!
```

UDP method:

Server:

```
import java.io.*;
import java.net.*;
public class Server {
        public static void main(String[] args) throws Exception
        {
                DatagramSocket socket=new DatagramSocket(9861);
                byte receiveByte[]=new byte[1024];
                byte sendByte[]=new byte[1024];
                while(true)
```

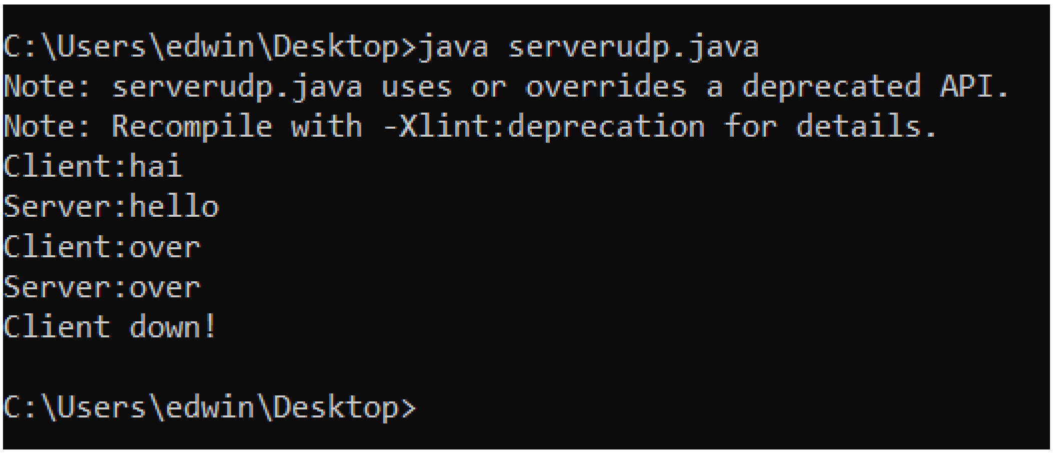
```
                    {
                        DatagramPacket receivePacket=new
DatagramPacket(receiveByte,receiveByte.length);
                        socket.receive(receivePacket);
                        String receiveStr=new String(receivePacket.getData());
                        receiveStr=receiveStr.trim();
                        System.out.println("Client:"+receiveStr);
                        DataInputStream din=new DataInputStream(System.in);
                        System.out.print("Server:");
                        String sendStr=din.readLine();
                        sendByte=sendStr.getBytes();
                        InetAddress ip=receivePacket.getAddress();
                        int port=receivePacket.getPort();
                        DatagramPacket sendPacket=new
DatagramPacket(sendByte,sendByte.length,ip,port);
                        socket.send(sendPacket);
                        if(receiveStr.equals("over"))
                        {
                            System.out.println("Client down!");
                            try{
                                socket.close();
                                break;
                            }
                            catch (Exception e){
                            System.out.println("Socket Closed :(");
                    }
                    }
                    }
          }
}
```

Client:

```
import java.net.*;
import java.io.*;
public class Client {
        public static void main(String[] args) throws Exception
        {
                DatagramSocket socket=new DatagramSocket();
                InetAddress ip=InetAddress.getLocalHost();
                byte sendByte[]=new byte[1024];
                byte receiveByte[]=new byte[1024];
                while(true)
                {
                    DataInputStream din=new DataInputStream(System.in);
                    System.out.print("Client:");
                    String sendStr=din.readLine();
                    sendByte=sendStr.getBytes();
                    DatagramPacket sendPacket=new
DatagramPacket(sendByte,sendByte.length,ip,9861);
                    socket.send(sendPacket);
                    DatagramPacket receivePacket=new
DatagramPacket(receiveByte,receiveByte.length);
                    socket.receive(receivePacket);
                    String receiveStr=new String(receivePacket.getData());
                    receiveStr=receiveStr.trim();
                        System.out.println("Server:"+receiveStr);
                        if(sendStr.equals("over"))
                    {
                        System.out.println("Client down!");
                        try{
                            socket.close();
                            break;
                    }
                    catch (Exception e){
                    System.out.println("Socket Closed :(");
               }
               }
               }
               }
     }
```

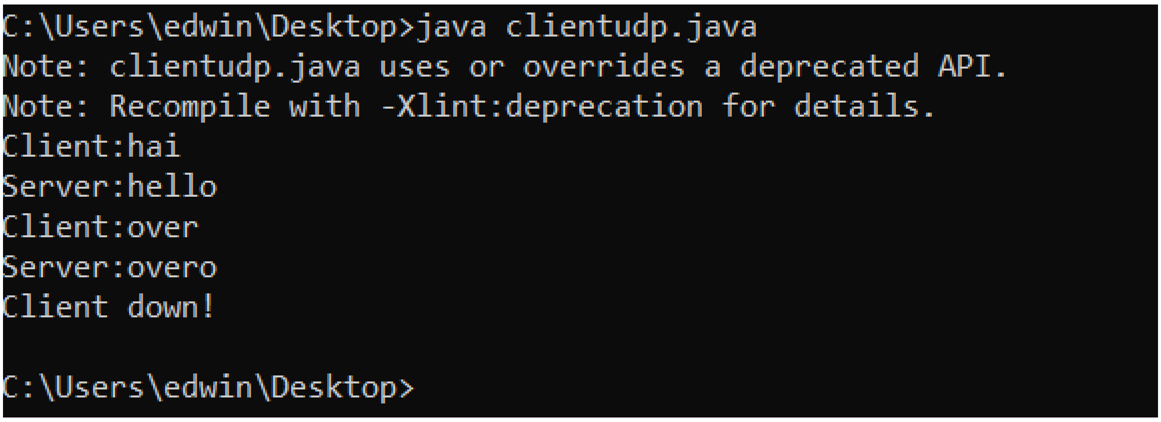
OUTPUT:

Server:



```
C:\Users\edwin\Desktop>java serverudp.java
Note: serverudp.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Client:hai
Server:hello
Client:over
Server:over
Client down!

C:\Users\edwin\Desktop>
```

Client:



```
C:\Users\edwin\Desktop>java clientudp.java
Note: clientudp.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Client:hai
Server:hello
Client:over
Server:overo
Client down!

C:\Users\edwin\Desktop>
```

RESULT:

Network communication using java is successfully achieved with tcp and udp methods