Problem Statement : Write a program using TCP sockets for wired network to implement
a. Peer to Peer Chat
b. Multiuser Chat

a. Peer to peer chat

Client

```
import java.util.*;
import java.net.*;
import java.io.*;
public class tcp_client
{
  public static void main(String[] args) throws Exception
  {
     Socket sock = new Socket("127.0.0.1", 3000);
                              // reading from keyboard (keyRead object)
     BufferedReader keyRead = new BufferedReader(new InputStreamReader(System.in));
                             // sending to client (pwrite object)
     OutputStream ostream = sock.getOutputStream();
     PrintWriter pwrite = new PrintWriter(ostream, true);

                             // receiving from server ( receiveRead  object)
     InputStream istream = sock.getInputStream();
     BufferedReader receiveRead = new BufferedReader(new InputStreamReader(istream));

     System.out.println("Start the chitchat, type and press Enter key");

     String receiveMessage, sendMessage;
     while(true)
     {
        sendMessage = keyRead.readLine();  // keyboard reading
        pwrite.println(sendMessage);       // sending to server
        pwrite.flush();                    // flush the data
        if((receiveMessage = receiveRead.readLine()) != null) //receive from server
        {
            System.out.println(receiveMessage); // displaying at DOS prompt
        }
      }
    }
}
```

Server :

```
import java.io.BufferedReader;

public class tcp_server
{
  public tcp_server() {}

  public static void main(String[] paramArrayOfString) throws Exception {
    java.net.ServerSocket localServerSocket = new java.net.ServerSocket(3000);
    System.out.println("Server  ready for chatting");
    java.net.Socket localSocket = localServerSocket.accept();

    BufferedReader localBufferedReader1 = new BufferedReader(new
java.io.InputStreamReader(System.in));

    java.io.OutputStream localOutputStream = localSocket.getOutputStream();
    java.io.PrintWriter localPrintWriter = new java.io.PrintWriter(localOutputStream, true);


    java.io.InputStream localInputStream = localSocket.getInputStream();
    BufferedReader localBufferedReader2 = new BufferedReader(new
java.io.InputStreamReader(localInputStream));

    for (;;)
    {
      String str1;
      if ((str1 = localBufferedReader2.readLine()) != null)
      {
        System.out.println(str1);
      }
      String str2 = localBufferedReader1.readLine();
      localPrintWriter.println(str2);
      localPrintWriter.flush();
    }
  }
}
```

Output :

Server :

```
G:\CO5G\CN\peer to peer chat>javac tcp_server.java

G:\CO5G\CN\peer to peer chat>java tcp_server
Server  ready for chatting
Hello
How are you
I m fine testing programs now
```

Client :

```
G:\CO5G\CN\peer to peer chat>javac tcp_client.java

G:\CO5G\CN\peer to peer chat>java tcp_client
Start the chitchat, type and press Enter key
Hello
How are you
I m fine testing programs now
```

Multi User Chat:

Server :

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.HashSet;


public class ChatServer {

    private static final int PORT = 9001;

    private static HashSet<String> names = new HashSet<String>();

    private static HashSet<PrintWriter> writers = new HashSet<PrintWriter>();

    public static void main(String[] args) throws Exception {
        System.out.println("The chat server is running.");
        ServerSocket listener = new ServerSocket(PORT);
        try {
            while (true) {
                new Handler(listener.accept()).start();
            }
        } finally {
            listener.close();
        }
    }

    private static class Handler extends Thread {
        private String name;
        private Socket socket;
        private BufferedReader in;
        private PrintWriter out;

        public Handler(Socket socket) {
            this.socket = socket;
        }

        public void run() {
            try {

                in = new BufferedReader(new InputStreamReader(
                    socket.getInputStream()));
                out = new PrintWriter(socket.getOutputStream(), true);

                while (true) {
                    out.println("SUBMITNAME");
                    name = in.readLine();
                    if (name == null) {
                        return;
                    }
                    synchronized (names) {
                        if (!names.contains(name)) {
                            names.add(name);
                            break;
                        }
                    }
                }

                out.println("NAMEACCEPTED");
                writers.add(out);

                while (true) {
                    String input = in.readLine();
                    if (input == null) {
                        return;
                    }
                    for (PrintWriter writer : writers) {
                        writer.println("MESSAGE " + name + ": " + input);
                    }
                }
            } catch (IOException e) {
                System.out.println(e);
            } finally {

                if (name != null) {
                    names.remove(name);
                }
                if (out != null) {
                    writers.remove(out);
                }
                try {
                    socket.close();
                } catch (IOException e) {
                }
            }
        }
    }
}
```

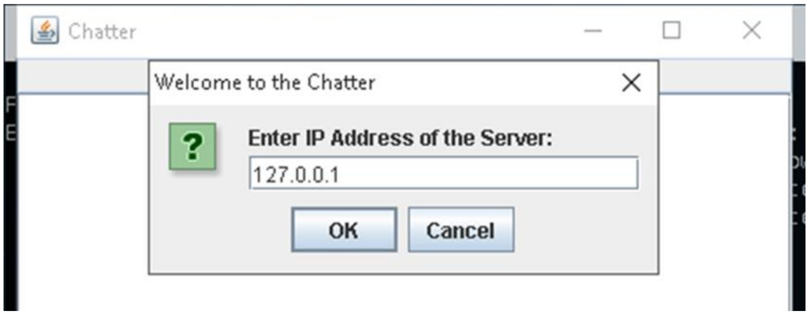
Client :

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
```

```java
public class ChatClient {
    BufferedReader in;
    PrintWriter out;
    JFrame frame = new JFrame("Chatter");
    JTextField textField = new JTextField(40);
    JTextArea messageArea = new JTextArea(8, 40);

    public ChatClient() {
        textField.setEditable(false);
        messageArea.setEditable(false);
        frame.getContentPane().add(textField, "North");
        frame.getContentPane().add(new JScrollPane(messageArea), "Center");
        frame.pack();

        textField.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                out.println(textField.getText());
                textField.setText("");
            }
        });
    }

    private String getServerAddress() {
        return JOptionPane.showInputDialog(
            frame,
            "Enter IP Address of the Server:",
            "Welcome to the Chatter",
            JOptionPane.QUESTION_MESSAGE);
    }

    private String getName() {
        return JOptionPane.showInputDialog(
            frame,
            "Enter the name:",
            "Screen name selection",
            JOptionPane.PLAIN_MESSAGE);
    }

    private void run() throws IOException {
        String serverAddress = getServerAddress();
        Socket socket = new Socket(serverAddress, 9001);
        in = new BufferedReader(new InputStreamReader(
            socket.getInputStream()));
        out = new PrintWriter(socket.getOutputStream(), true);

        while (true) {
            String line = in.readLine();
            if (line.startsWith("SUBMITNAME")) {
                out.println(getName());
            } else if (line.startsWith("NAMEACCEPTED")) {
                textField.setEditable(true);
            } else if (line.startsWith("MESSAGE")) {
                messageArea.append(line.substring(8) + "\n");
            }
        }
    }
    public static void main(String[] args) throws Exception {
        ChatClient client = new ChatClient();
        client.frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        client.frame.setVisible(true);
        client.run();
    }
}
```

Output :

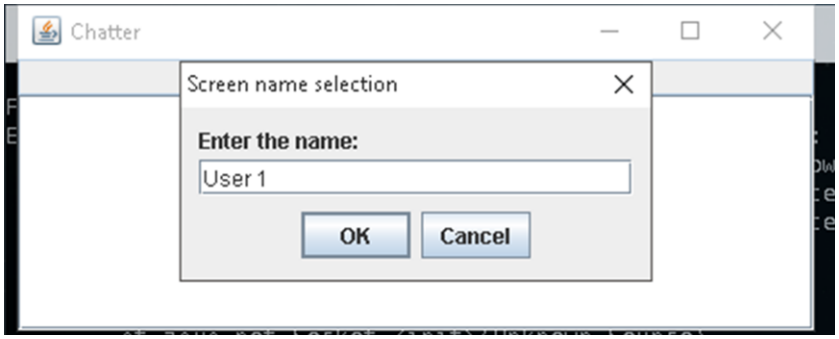
```
F:\CO5G\CNL>java ChatServer
The chat server is running.
```



Welcome to the Chatter

Enter IP Address of the Server:

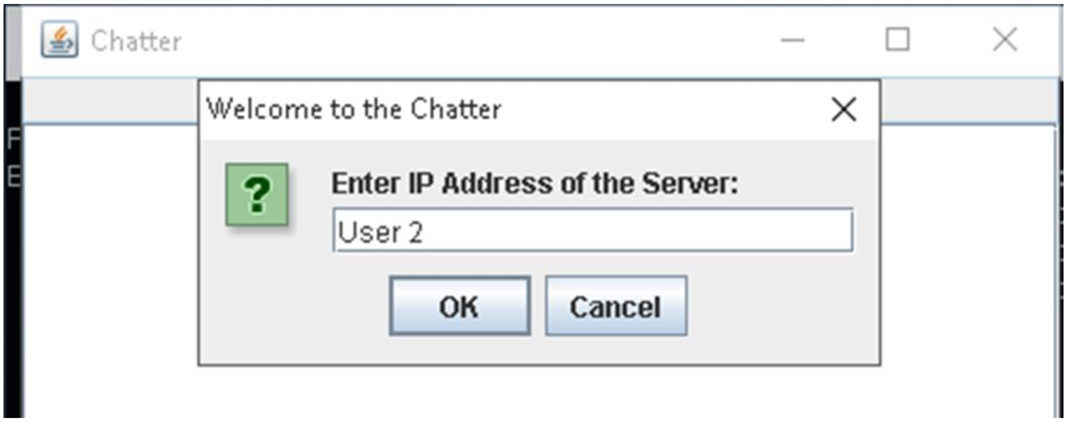
127.0.0.1



Screen name selection

Enter the name:

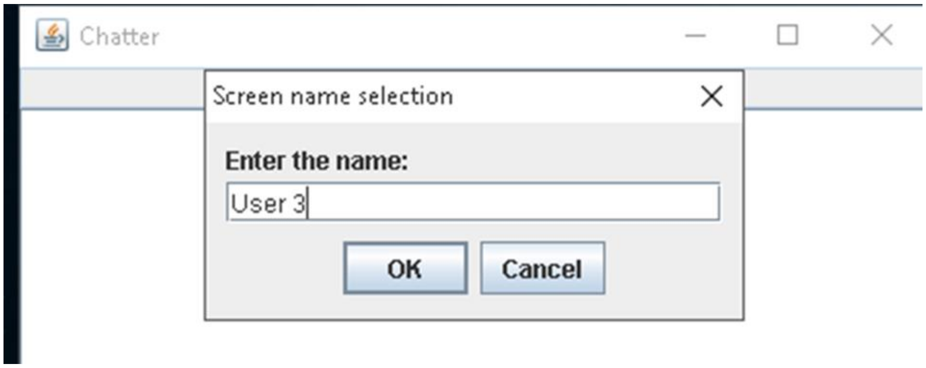
User 1



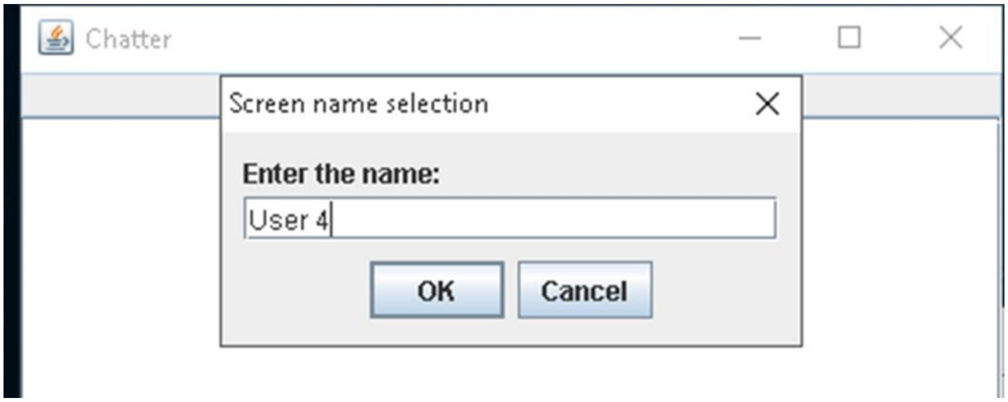
Welcome to the Chatter

Enter IP Address of the Server:

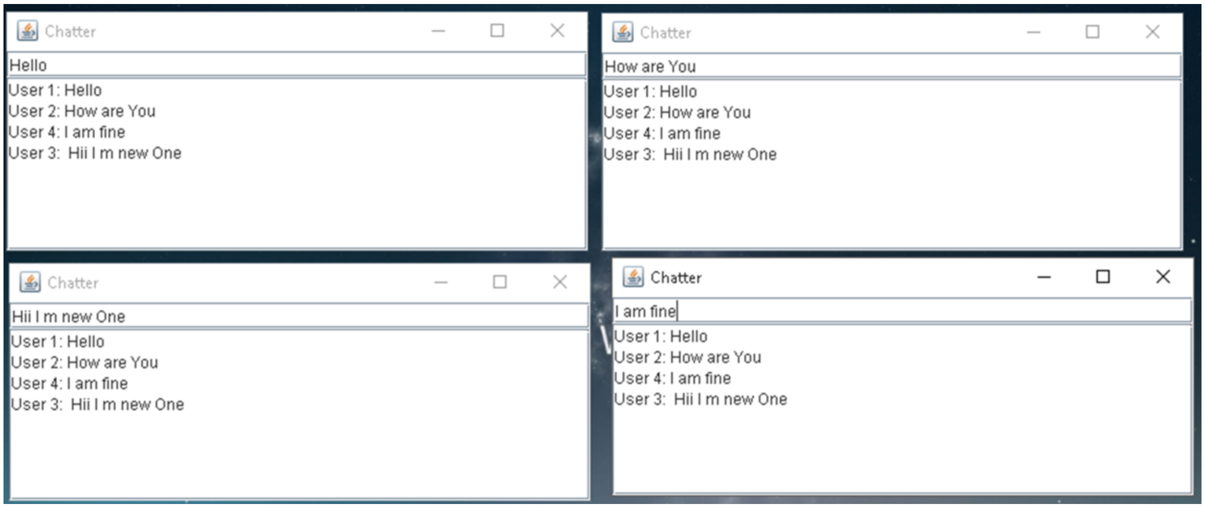
User 2



Enter the name: User 3



Enter the name: User 4



```
Hello
User 1: Hello
User 2: How are You
User 4: I am fine
User 3:  Hii I m new One
```

```
How are You
User 1: Hello
User 2: How are You
User 4: I am fine
User 3:  Hii I m new One
```

```
Hii I m new One
User 1: Hello
User 2: How are You
User 4: I am fine
User 3:  Hii I m new One
```

```
I am fine
User 1: Hello
User 2: How are You
User 4: I am fine
User 3:  Hii I m new One
```