```
import java.io.*;
import javax.swing.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;

public class chatkaro12mClient extends Panel
{
   TextArea receivedText;
   Socket sock;
   private GridBagConstraints c;
   private GridBagLayout gridBag;
   private Frame frame;
   private Label label;
   JButton send;
   JButton exit;
   private int port=5001;
   private TextArea sendText;
   private String hostname;
   private String username;
   private DataOutputStream remoteOut;
   ImageIcon i1,i2;

   public static void main(String args[])
   {
      if(args.length != 2)
      {
         System.out.println("Format is : java chatkaro12mClient  ");
         return;
      }

      Frame f1=new Frame("Welcome Protocol");
      f1.resize(800,600);
      f1.show();
      JOptionPane.showMessageDialog(f1,
         "Welcome " + args[0] + ".. Have a nice Session.","Welcome",0);
      Frame f= new Frame("Connecting to Mr. "+args[0]);
      chatkaro12mClient chat=new chatkaro12mClient(f,args[0],args[1]);
      f.add("Center",chat);
      f.setSize(800,600);
      f.show();
      chat.client();
   }

   public chatkaro12mClient(Frame f,String user,String host)
   {
      frame=f;
      frame.addWindowListener(new WindowExitHandler());
      username=user;
      hostname=host;
      label=new Label("Text to send :");

      add(label);
      sendText=new TextArea(15,30);
      add(sendText);
      label= new Label("Text received :");
      add(label);
      receivedText=new TextArea(15,30);
      add(receivedText);

      ImageIcon i1=new ImageIcon("click.gif");
      ImageIcon i2=new ImageIcon("doorin2.gif");
      send=new JButton(i1);
```

```
   64:       exit=new JButton(i2);
   65:       add(send);
   66:       add(exit);
   67:       send.addActionListener(new TextActionHandler());
   68:       exit.addActionListener(new EXIT());
   69:    }
   70:
   71:    void client()
   72:    {
   73:       try
   74:       {
   75:          if(hostname.equals("local"))
   76:             hostname=null;
   77:
   78:          InetAddress serverAddr= InetAddress.getByName(hostname);
   79:          sock=new Socket(serverAddr.getHostName(),port);
   80:          remoteOut=new DataOutputStream(sock.getOutputStream());
   81:          System.out.println("Connected to server "
   82:             + serverAddr.getHostName() + " on port " + sock.getPort());
   83:          new chatkaro12mClientReceive(this).start();
   84:       }
   85:       catch(IOException e)
   86:       {
   87:          System.out.println(e.getMessage() +
   88:             " : Failed to connect to server.");
   89:       }
   90:    }
   91:
   92:    protected void finalize() throws Throwable
   93:    {
   94:       try
   95:       {
   96:          if(remoteOut != null)
   97:             remoteOut.close();
   98:          if(sock != null)
   99:             sock.close();
  100:       }
  101:       catch(IOException x)
  102:       {
  103:       }
  104:       super.finalize();
  105:    }
  106:
  107:    class WindowExitHandler extends WindowAdapter
  108:    {
  109:       public void windowClosing(WindowEvent e)
  110:       {
  111:          Window w=e.getWindow();
  112:          w.setVisible(false);
  113:          w.dispose();
  114:          System.exit(0);
  115:       }
  116:    }
  117:
  118:    class EXIT implements ActionListener
  119:    {
  120:       public void actionPerformed(ActionEvent e)
  121:       {
  122:          if((JOptionPane.showConfirmDialog(new Frame(),
  123:             "Are You Sure to close the Session?"))==JOptionPane.YES_OPTION)
  124:          {
  125:             JOptionPane.showMessageDialog(new Frame(),
  126:                "Thank U. Visit Again. ","Good Bye",0);
```

```
  127:             System.exit(0);
  128:          }
  129:       }
  130:    }
  131:
  132:    class TextActionHandler implements ActionListener
  133:    {
  134:       public void actionPerformed(ActionEvent e)
  135:       {
  136:          try
  137:          {
  138:             remoteOut.writeUTF(sendText.getText());
  139:             JOptionPane.showMessageDialog(new Frame(),
  140:                "Your msg has been sent ","Confirmation",0);
  141:             sendText.setText("");
  142:          }
  143:          catch(IOException x)
  144:          {
  145:             System.out.println(x.getMessage() + " : connection to peer lost.");
  146:          }
  147:       }
  148:    }
  149: }
  150:
  151: class chatkaro12mClientReceive extends Thread
  152: {
  153:    private chatkaro12mClient chat;
  154:
  155:    chatkaro12mClientReceive(chatkaro12mClient chat)
  156:    {
  157:       this.chat=chat;
  158:    }
  159:
  160:    public synchronized void run()
  161:    {
  162:       String s;
  163:       DataInputStream remoteIn=null;
  164:       try
  165:       {
  166:          remoteIn= new DataInputStream(chat.sock.getInputStream());
  167:          while(true)
  168:          {
  169:             s=remoteIn.readUTF();
  170:             chat.receivedText.setText(s);
  171:          }
  172:       }
  173:       catch(IOException e)
  174:       {
  175:          System.out.println(e.getMessage() + " : connection to peer lost.");
  176:       }
  177:       finally
  178:       {
  179:          try
  180:          {
  181:             if(remoteIn !=null)
  182:                remoteIn.close();
  183:          }
  184:          catch(IOException x)
  185:          {
  186:          }
  187:       }
  188:    }
  189: }
```