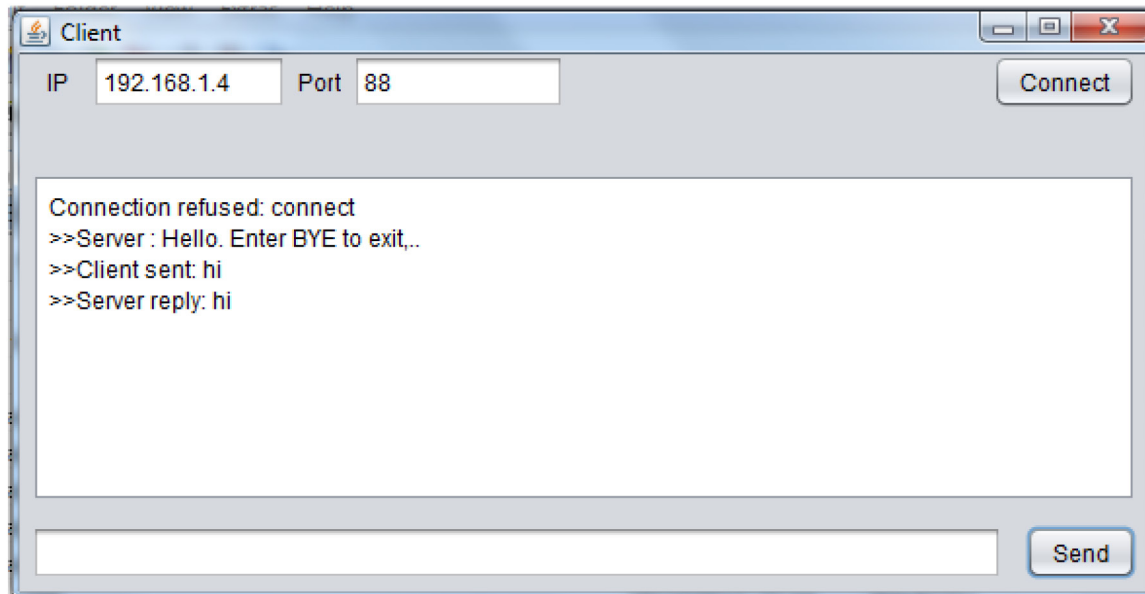


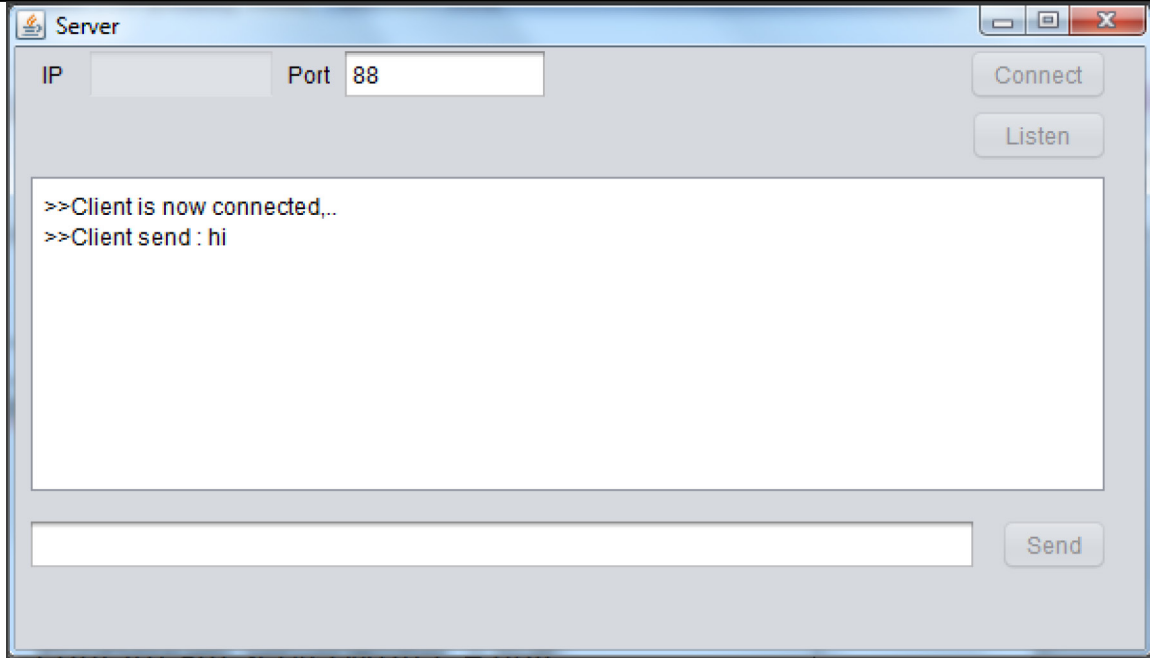
Practical Applications in CS I - Session 04.02



Client	Class declaration	<pre>Socket clientSocket = null; PrintWriter clientWriter = null; BufferedReader clientReader = null; String ip; int port;</pre>
	Connect button	<pre>ip = jTextField2.getText(); port = Integer.parseInt(jTextField3.getText()); try { clientSocket = new Socket(ip, port); clientWriter = new PrintWriter(clientSocket.getOutputStream(), true); clientReader = new BufferedReader(new InputStreamReader(clientSocket.getInputStream ())); jTextArea1.append(clientReader.readLine()+"\n") ;</pre>

		<pre> } catch (Exception e) { JTextArea1.append(e.getMessage()+"\n"); } </pre>
	Send button	<pre> try { clientWriter.println(jTextField1.getText()); JTextArea1.append(">>Client sent: " + jTextField1.getText()+"\n"); JTextArea1.append(">>Server reply: " + clientReader.readLine()+"\n"); } catch (Exception e) { JTextArea1.append(e.getMessage()+"\n"); } finally { if (jTextField1.getText().equalsIgnoreCase("BYE")) { try { if (clientWriter != null) { clientWriter.close(); } if (clientReader != null) { clientReader.close(); } if (clientSocket != null) { clientSocket.close(); } JTextArea1.append(">>Client: Socket closed.\n"); } catch (Exception e) { JTextArea1.append(e.getMessage()+"\n"); } } } </pre>

		} jTextField1.setText("");
--	--	-------------------------------



Server	Class declaration	<pre>int port; ServerSocket ss = null; Timer t1; Timer t2; BufferedReader serverReader = null; PrintStream serverWriter = null; Socket incoming = null;</pre>
	initSocket	<pre>void initSocket() { port = Integer.parseInt(jTextField3.getText()); try { ss = new ServerSocket(port); } catch (IOException e) { jTextArea1.append(e.getMessage() + "\n"); } try { incoming = ss.accept();</pre>

		<pre> } catch (IOException e) { } try { incoming.setSoTimeout(60000); //60 seconds } catch (SocketException e) { } try { serverReader = new BufferedReader(new InputStreamReader(incoming.getInputStream())); serverWriter = new PrintStream(incoming.getOutputStream()); serverWriter.println(">>Server : Hello. Enter BYE to exit,.."); jTextArea1.append(">>Client is now connected,..\\n"); //readSocket(); t1.stop(); t2.start(); jButton3.setText("Listen"); jButton3.setEnabled(false); } catch (Exception e) { jTextArea1.append(e.getMessage() + "\\n"); } } </pre>
	readSocket	<pre> void readSocket() { try { String str = serverReader.readLine(); </pre>

		<pre> serverWriter.println(str); jTextArea1.append(">>Client send : " + str + "\n"); if (str.equalsIgnoreCase("BYE")) { t2.stop(); jTextArea1.append(">> Cleint is now dis-connected,..\\n"); incoming.close(); ss.close(); jButton3.setEnabled(true); } } catch (Exception e) { jTextArea1.append(e.getMessage()+"\\n"); } } </pre>
	Class constructo r	<pre> t1 = new Timer(1000, new ActionListener() { @Override public void actionPerformed(ActionEvent e) { initSocket(); } }); t2 = new Timer(1000, new ActionListener() { @Override public void actionPerformed(ActionEvent e) { readSocket(); } }); </pre>
	Listen button	<pre> if (jButton3.getText().equalsIgnoreCase("Listen")) { </pre>

		<pre>t1.start(); jButton3.setText("Stop"); } else { t1.stop(); jButton3.setText("Listen"); }</pre>
--	--	--