**Assignment 2 Pre-submission check list**

**General Testing requirements – Applicable to all parts**

Code has been tested on virtual machines in EN Lab

Code has been tested on real machine in EN Lab

**Comments – Applicable to all parts**

Each Class has a comment at the top that includes the description of

what it does

your name and ID.

Each method has a header describing what it does, and includes @param, @return and @exception

There are comments throughout the code explaining how it works

All comments follow javadoc requirements

**References**

All code that has been adapted from or directly copied from other sources must clearly show the source. This should be done at the top of any method that it applies to.

Example 1

// this code has been adapted from code found at this source

//<http://www.youtube.com/watch?v=Q0ue-T0Z6Zs>, accessed 27 April, 2014

Example 2

// This method was copied from (adapted from ) lecture 3, slide 27.

**General Submission requirements**

Code must be submitted in zip file to ESP

Code shown in class must be the same as code submitted to ESP

**Checks for Part A - Hello Server**

\_\_\_ Has been written using UDP

The server, once started

\_\_\_ runs in a loop and

\_\_\_ can respond to multiple different requests.

When the client program is run

\_\_\_ IP and the port of the server are entered at the command line

**For example, java ClientProgram 127.0.0.1 4001**

\_\_\_ IP and port number are validated (poor input will not crash program)

\_\_\_ The client receives your ID and Name and displays it on the console.

\_\_\_ Other clients, on the same machine, or other machines may also make the request and get a response.

**Marking**

\_\_\_ Code works as expected (all above working)

\_\_\_ Student understands code

\_\_\_ Code documentation is thorough and makes sense

5, if all above ok, otherwise 0 marks.

Your mark for Part A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / 5

**Check for, Part B - File Server**

Server Part

\_\_\_ Has been written using TCP \*\*

\_\_\_ (1) Has the ability to work with multiple clients without being restarted

When file name is received

\_\_\_ (1) validates file request string without crashing

\_\_\_ (1) locates file

\_\_\_ (1) sends file back to client

\_\_\_ (1) closes connection

Client Part

\_\_\_ (1) is started at the command line at time program starts

\_\_\_ (1) receives IP, port and filename at command line

\_\_\_ (1) validates strings received at command line

\_\_\_ (1) connects to server

\_\_\_ (1) makes request and receives file.

\_\_\_ (1) stores received file in local folder

\_\_\_ (3) Test 1 – small text file \*\*

\_\_\_ (3) Test 2 – mp3 file (-5)

\_\_\_ (3) Test 3 – large image (-5)

**Marking**

\*\* 0 MARKS IF THIS NOT DONE

\_\_\_ Student understands and can explain code (0/10 if not understood)

\_\_\_ Code documentation is thorough and makes sense (0/10 poor)

Your mark for Part B (above divided by 2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / 10

**Checks for , Part C Chat Program - TCP or UDP**

**Testing**

Deploy software on at least four machines (may be virtual or real network)

\_\_\_ \*\* The software on each machine is the same and there is no central server

\_\_\_ (2) Peer list is read from file

\_\_\_ (2) Peer list is displayed correctly on screen

In the format “Peer 1, Fred <IP ……>”

Type message on one machine and view display on other computers

On sending Machine

\_\_\_ (1) is displayed , \_\_\_ (1) is displayed correctly

Buzz <IP 123.212.132> Hi…”

On receiving Machines

\_\_\_ (2) is displayed , \_\_\_ (2) is displayed correctly

\_\_\_ (3) Chat is exchanged between clients without error and in the manner expected

Run software on machine without approved IP

\_\_\_ (1) Rejected message is displayed correctly

“Unauthorized chat request from <IP 123.222.132.111>”

\_\_\_ (1) No message displayed on second attempt

\_\_ \*\* Only one command line window used for all interactions

\_\_ \*\* The student understands the code and can explain the code

\_\_ (-5) The code is carefully and meticulously documented

0 marks for part C if \*\* items not satisfactory

Your mark for Part B \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / 15

**Part D - Bonus (5 marks, if chat messages are encrypted)**

Testing

\_\_\_ understands and can explain where, how and when code is encrypted

\_\_\_ understands and can explain how keys are used

\_\_\_ code is actually encrypted (show by capturing a frame using Wireshark)

\_\_\_ all code is documented

5 marks if all above ok, otherwise 0 marks.

Your mark for Part D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / 5