Swinburne University of Technology
Faculty of Information and Communication Technologies
Data Communications & Security
6% - (Total of 30 marks) - Individual

Submission Requirements:

- **To be submitted to ESP by 5pm on Friday May 2.**
  https://esp.ict.swin.edu.au/
- **Submission must include source code for each part, with very detailed comments about how it works.**
- **To be demonstrated to your tutor in week 9 in your lab class.**

Projects

All projects are console applications
All projects must be written in Java.

1. Hello Server – UDP (5 Marks)
2. File Server – TCP (10 marks)
3. Chat Program – (15 marks)
4. Bonus (5 marks, if chat messages are encrypted)

Qualities

This is a code writing assignment and it is expected that a significant amount of the code you write will be sourced from the work of others. The work sourced from others MUST BE acknowledged by the use of inline comments in your source code.

You may not submit other students work. You may not ask someone to write part or all of the code or pay someone to write part or all of the code.

The main criteria to pass are;

1. You develop working programs
2. You provide code that is meticulously documented
   a. Uses javadoc
   b. Every line is explained in detail
   c. Includes references to all sources
3. You understand the work submitted and can explain how it works to your tutor
4. You successfully demonstrate the code in class to your tutor.
5. All projects run on both the virtual and the real networks in EN305.***Hello Server – UDP***

Hello Server – UDP

Server Part

- Runs in a loop
- When a request comes in it returns “Hello, my name is ……. And my Id is ……”

Client Part

- Receives IP and port at command line
- Connects to server
- Makes request
- Displays response on screen

File Server - TCP

Server Part

- Runs in a loop
- Receives request in the format “Send myfile.txt”
- Sends file to client
- Closes connection
- Waits for another request

Client Part

- Receives IP, port and filename at command line
- Connects to server
- Makes request and receives file.
- Stores file in local folder

Notes:

Must handle large files
Must handle files other than text files.

Chat Program - TCP or UDP

- Peer to peer architecture
- Once software runs
    - A list of valid peers are read from file
    - The peers are displayed on the screen, one per line
        - In the format “Peer 1, Fred <IP ……>”
    - Any message typed at the screen is sent to all peers
    - Any message received from a peer should be displayed on the screen
    - Messages displayed should be in the format
        - “Buzz <IP 123.212.132> Hi…”
    - Both sent and received messages should be displayed
    - Unauthorized contacts should result in the following message being displayed (once only per IP)
        - “Unauthorized chat request from <IP 123.222.132.111>”
    - All in and out going messages should be in one command window
    - If a message from unauthorized chatter arrives it should be rejected