

# CISC 4615 Data Communication & Networks

## Lab 1: Socket Programming (Due: 09/23)

---

### Lab additional instructions

You completed lab1 is expected to have the following functionalities.

1. The server can accept several different commands:
    - `list` : The server sends back all the active client IDs.
    - `msg ID string` : The server should be able to understand that this client wants to send the `msg(string)` to the other client with the ID that listed the command. The server should be able to forward the message to the target as the following format `source ID: message_content`
    - `history ID` : The server should send back the chatting history between the requested client and the client with ID listed in the command.
    - `exit` : The The server should send back a message "Goodbye" and close the connection.
  2. The client should be able to communicate with both the server and other active clients.
- 

### System design ideas

The server is keep listing with the master socket. Whenever the client connect to the server, this client will be assign a new socket ID and the new ID will be add into an array.

Whenever the server receives anything from the client, it will take the following steps,

1. CheckMsg (String msg, int ID, char\* buffer....)
  - String msg: is the message that received from the clients
  - int ID: is the the client's socket ID
  - char\* buffer: is the pointer that points to the buffer to be sent to client.
  - Hint: you can use **strtok** function in c language.
  - Check the examples: [link](#).
2. List (int ID, char\* buffer....)

- int ID: is the ID comes from CheckMsg function.
- char\* buffer: is the pointer from CheckMsg function.
- This function should construct the buffer that contains all the active socket IDs.
- Hint: you may need to think about how the access the socket array, which is defined in main function (passing parameters by pointers/references).

### 3. SentMsg(string content, int source, int destination....)

- string content: is the message content (not the "command") that you extract in the CheckMsg function.
- int source / int destination: are the socket IDs that you obtained in the CheckMsg function.
- This function should store the message content first and then, send it based on the destination ID (it can also ask the main function to send it out to destination)

### 4. History(int source, int destination...)

- int source / int destination: are the socket IDs that you obtained in the CheckMsg function.
- You need to contruction the buffer that contains all the messages communication between source and destination, which you stored in function 3 or in main function.

The above functions are design ideas and the input parameters may not enough depending your implementation. Please think twice about the system design before you start coding.