Course:	**INFO6016 - Network Programming, Fall 2023**
Project # 1:	**TCP Client & Server, Chat Program**
Due Date:	**October 22, 2023, 11:59pm (2.5 weeks)**
Weight:	**15%**

Note: This project *can* be done in a group of 2

Description and Purpose

Your job is to create two separate programs. A C++ Chat Client that communicates through TCP and a C++ Chat Server that communicates through TCP. *You may create either a GUI or command line interface for your chat client.*

Details

Write a **Buffer Class** as shown in class. This class must implement the following:

- Initialize with size N
- **Grow** when serializing past the write index
- Serialize, Deserialize unsigned int (32 bit)
- Serialize, Deserialize unsigned short (16 bit)
- Serialize, Deserialize string

Create a **Protocol** as discussed in class. The protocol must follow the following rules:

- Must use length prefix message framing
- Big Endian must be used for the protocol for the number values.
    - *Note: String does not need to have its endian swapped, it is just a sequence of chars.*

Develop a C++ **Server** using BSD Sockets. The server must include the following:

- Handle connections and messages without blocking must be concurrent.
    - Concurrency can be done in the following ways:
        - 1 Thread per Connection
        - Non-blocking using `select();`
- Deserialize messages properly
- Ability for connections to join a room
- Ability for connections to leave a room

- Ability for connections to send a message to a room and broadcast to peers
    - **[name]** `hello everyone!`
- Ability for connections to join multiple rooms simultaneously
- When a connection joins a room, the server should broadcast:
    - **[name]** `has joined the room.`
- When a connection leaves a room, the server should broadcast:
    - **[name]** `has left the room`

Develop a C++ **Client** using BSD Sockets. The client must support the following actions

- Able to join a room.
- Able to leave a room.
- Able to send a message to a room
- Able to receive messages from a room

Important note: Do not forget to include all dependencies within your project, you might want to use your project directory for includes and libs.
i.e.
$(ProjectDir)includes
$(ProjectDir)lib

Plagiarism

➢ While you may freely "borrow" mine (or anyone else's) code, ***but*** your code should be "sufficiently" different from mine.

➢ In other words, you *cannot* simply use an existing game engine (or part of a game engine) to complete this assignment; it should be either completely new or **significantly modified**.

Grading Scheme

1. 15% marks will be deducted for each day late the project is submitted.

2. If your code does not compile, I will not mark it. This will get you a mark of zero (0).

3. If your code does not build (i.e. linker error) and run (i.e. no crazy run-time crash that is unexpected), I may investigate this further, but only if there is some simple problem and/or very slight and/or very obvious (and easy to fix) configuration error.

Item	Marks
Buffer Class 1. Initialize with size N (**1 mark**) 2. Grow (**2 marks**) 3. Serialize, Deserialize *type* (**2 marks *each**)	9
Protocol 1. Binary (**1 mark**) 2. Length Prefix message framing (**2 marks**) 3. Big Endian (**1 mark**)	4
Server 1. Concurrency (**5 marks**) 2. Deserialize Messages Properly (**10 marks**) 3. Connection can Join a room (**1 mark**) 4. Connection can Leave a room (**1 mark**) 5. Connection can send a message to a room (**2 marks**) 6. Connection can Join Multiple Rooms (**1 mark**) 7. Broadcast a connection joined a room (**1 mark**) 8. Broadcast a connection left a room (**1 mark**)	22
Client 1. Join room (**1 mark**) 2. Leave room (**1 mark**) 3. Send Message to room (**1 mark**) 4. Receive messages from a room (**1 mark**)	4
Source Control 1. Git (**3 marks**)	3
Total	42

Project Submission

The following are **required** for submitting your project:

- ReadMe.txt
  - Describe how to build your project.
  - Describe how to run your project.
  - Describe the user input options.
  -

- ➢ Git
    - If you prefer to submit your project through Github you may. You just need to send me a link to your github repository.

The following are **optional** for submitting your project:

- ➢ Video.mp4
    - Upload a video of you demoing your project.
    - You may include the video with your project submission or upload to YouTube and provide a link.
    - This helps start your portfolio

Project Corrections

If any corrections or changes are necessary they will be posted to the course web site and you will be notified of any changes in class. It is your responsibility to check the site periodically for changes to the project. Additional resources relating to the project may also be posted.