

CSCE 416 Programming Assignment 2

Group Chat

In this assignment, you will implement a **group chat server** that enables conversation between multiple **group chat clients**. Clients connect to the server and all the connected clients receive all the messages transmitted by any of the clients. This group communication is facilitated by the group chat server. A client reads the text typed by the user and sends it to the server. It is the responsibility of the server to relay that text to all the other clients. Clients receive the text relayed by the server and display it on the screen.

You need to write two programs: **GroupChatServer** and **GroupChatClient** using sockets. The functionality of each of these programs is described below.

GroupChatServer

The server's job is to create a server socket and wait for new clients or messages from existing clients. If a new client is connected, that client is added to the list of active clients. If a null message is received from an existing client (which happens when a client exits), that client is removed from the list of active clients.

To simultaneously watch for new clients and messages from multiple existing clients, you may use **threads** in 'java' or **select** method in 'python' or 'c'. You can find example usage of these approaches in TwoWayAsyncMesg.zip posted on the [Blackboard](#). Once you unzip it, you will find three folders, 'c', 'java', and 'python', corresponding to an implementation of the two-way messaging application in that programming language. Each folder contains two programs, TwoWayAsyncMesgServer and TwoWayAsyncMesgClient. The instructions on how to run them are given in the [README.txt](#) file in each folder.

GroupChatClient

The client program first establishes a connection to the server at the given host and port and then sends the given name. Thereafter, the client's job is to wait for either user input from the keyboard or messages from the server. Inputs from the user are sent to the server and the messages from the server are displayed on the screen. To implement this, you only need to make a minor change to the TwoWayAsyncMesgClient code.

Submission

- (1) Create a zip file containing the **GroupChatServer** and **GroupChatClient** source files
- (2) Rename the zip file as YOURLASTNAME_P2.zip (YOURLASTNAME in all caps);
- (3) Upload it in the [Blackboard](#).

Grade Breakdown

- (1) 10%: You submitted your assignment on-time and the files compile correctly.
- (2) 30%: Client code works correctly in sending its name and exchanging messages with server.
- (3) 30%: Server works correctly in relaying messages between clients.
- (4) 30%: Server works correctly in handling clients leaving and joining the group chat.