

**Lab #12**  
**CSCI 201**

Title

Sequential Chatting

Lecture Topics Emphasized

Networking

Monitors

Locks and Conditions

Introduction

Chatting is a very common application. We wrote a chat server and chat client in class, but we are going to add some features to the chat application to force specific rules to be followed. We will make sure that clients communicate in a specific order. Multiple messages can be sent from a user, but the messages must be delivered in a given user order.

Description

For this lab, you will modify the `ChatServer` and `ChatClient` programs that are posted for today's lab (or feel free to write them from scratch if you would like). Currently, a client can send a message to the server at any time, and the server broadcasts the message back out to all of the clients who are currently connected. This allows one client to send multiple messages before receiving a response from any of the other clients.

We are going to modify the program so that messages from clients will be delivered in a specific order (assuming they have sent a message). Instead of the server allowing a message to be broadcast immediately upon receipt, the server will only broadcast messages every one second. After a message has been broadcast, another message will not be broadcast until after at least one second has elapsed.

In addition to that, we want clients to broadcast messages based on a specific order. The order will be determined based on the order the clients have connected to the server. Let's take an example here. Assume four clients have connected to the server: client 1, client 2, client 3, and client 4. During the first one second period, only client 1 can send a message. Client 1 can send as many messages as he wants during that period, but as soon as one second elapses, only client 2 can send messages during the next second. This continues in a round-robin fashion until the time period for the last client has elapsed, which will return the time to client 1 again.

The solution to the above problem can be implemented with locks and conditions. The client code should not be affected. When the client sends a message to the server, the server will hold onto the message if it is not that client's turn to broadcast. To notify the client when it is their turn, you can use conditions on a lock. Signaling the proper client could be a method for letting them know they can broadcast.

There are multiple ways to solve this problem, but I want you to get some experience using locks and conditions.

### Grading Criteria

Labs are not graded based on any given criteria but are instead graded on effort and attendance. If you arrived to lab within the first 10 minutes and worked on it the for the entire duration of the lab, you will receive full credit regardless of whether you completed it. TAs will not grade labs until after at least half the lab period has elapsed. Use the lab time as an opportunity to more fully understand the course material and ask your TA questions.