Questions:

1. If student A has the server program running on machine P, could student B use his client program running on machine Q to connect to A's server program if student B knows the port on which student A's server is running?

   No, see question 3 for an explanation

2. If your answer to question (1) is yes, then explain how to prevent this from happening.

   n/a

3. If your answer to question (1) is no, then explain why this is not possible.

   With some basic testing and analysis, I found that this is not possible because the new thread only accepts a client once. When one client has joined, no other clients can connect to that thread/port.

4. Why do you need to use a port # > 1024?

   Ports below this require sudo privileges to bind to and are typically designated/reserved for a specific service. Above this number, there is more freedom to bind and host other things on the ports. I typically use the 8000s and 9000s for custom projects and private servers as there is a fair amount of freedom and wiggle room to

Feedback Questions:

Was this homework too difficult, or too easy?

> It was alright, maybe a little much, but it depends on if all the projects will be this involved. This was alright, but only if there aren't too many projects like this.

Was the assignment fun or challenging?

> It was both fun and challenging.

Was there something that was unclear?

> Several different things on the project spec doc could have been more clear. I have had some professors be very anal about every little detail and if the spacing is wrong, points would be removed. However this class seems to be much more realistic, and I am still trying to figure out how strict requirements are. Based on talking in class, it sounds like as long as we have the general program working within specs, we should be fine, and the smaller details like logging format are good as long as they are complete.

Was the homework too long for the given amount of time?

> It was kind of long but overall I was able to finish it. I did have to dedicate a lot of time to it, even though I have done something relatively similar in the past.

What did you learn from this homework?

> I learned about multithreading, multithreaded queues, writing a simple publisher/subscriber framework, blocking and non-blocking calls, and sockets in Python.

Test Cases:

> The program was tested manually by starting the server and connecting with one client. I would either disconnect or send 1 to many messages first. I repeated the tests with multiple clients at the same time as well, both joining and immediately leaving and force quitting, as well as sending messages or using the LIST command. Additional testing was done to allow either the client or the server to vanish while the other handles this gracefully.