Graba, Exercise 3.5 (Variation): Chat Room: GUI Client and Multi-Threaded Server:

Implement a basic chat-room application that employs a GUI client and a multi-threaded server:

- The client presents a GUI to the human user. The GUI initially has a dialog box asking the user for his/her nickname. After the user enters the nickname, the client establishes a socket connection to the server and sends the user’s name to the server through the socket. The GUI includes an input area where the user can enter one line messages and press a button which sends the message to the server. The client program watches the messages that the user is sending to the chat room. If the user enters the message: “Bye”, the client program will send this message to the server, and then close the connection to the server.

- The server listens for connections from clients. After establishing a connection with a client, the server receives the name (e.g., “Tom”) of the client. The server responds to this message by sending a message to all connected clients saying: “Tom has entered the chat room.”. Next the server receives a sequence of one-line messages. The server sends each message it receives to all connected clients, along with an indication of the sender’s name. At some point, the server may receive the message: “Bye”. It responds by sending to all connected clients the message: “Tom has left the chat room.”. Finally, the server closes the connection with the client.

I have supplied you with the GUI portion of the client program. You will need to implement the networking software that allows the client to interact with the server. You must implement the server program.

Extra Credit (10%): Implement a second version of the server using Java’s non-blocking IO facilities.

Testing:
1. Test your program with the client and server running on the same machine.
2. Test your program with the client and server running on different machines.

Administration: Students will develop and run their programs on the Linux machines in the Asprey lab. We need a mechanism for keeping students’ programs from interfering with each other. For this reason each student must use a distinct port number on which his/her client and server programs will communicate. Your port number will be derived from your Vassar post-office box number as follows:

Port Number = 50000 + Vassar Post Office Box Number