Why should there be a separate class to receive data from the server and print it, and the client only gets data from the user and sends it to the server. Also, why is the class called a 'listener'?

There should be a separate class to receive data from the server and print since without this class, it is not impossible that the client could get hung up on a message and be unable to send any messages. With separate classes, if an incoming message stops the message reception system, messages can still be sent out. It is called a listener since it "listens" for input from the server.

Explain why you need a separate thread for each client, and why you cannot handle all clients in the main server thread. Conceptually, why is the listener class 'ClientSideServerListener' different from the class 'ServerSideClientIO'?

The clients need their own threads so they can function simultaneously. If all clients were managed in the main thread then users would not be able to send messages at the same time as other users in the same main thread. With just a couple of users that's not necessarily going to be a problem, but when the project is scaled, given enough users, there will almost always be requests to send simultaneously.

Explain why the broadcast() and remove() methods are synchronized. You may find it easier to answer this question after completing all programming.

Broadcast and remove are synchronized since they modify the same list, and could get stalled on each other otherwise.

Discuss all new methods and new code in existing methods that you wrote to handle LISTUSERS.

The LISTUSERS is handled by implementing an arraylist of clients, which can be sent. This arraylist allows clients to be accessed.