My Networked Morse Code, i.e. Module 5, server/client applications already allows for any number of clients to connect, each of which could come and go as they please. Granted, I did set a hard limit in my server of 100 clients; however, this was just an arbitrary number and said requirement was not really necessary.

To allow for this functionality I primarily made usage of string functions, a closeConnection() method for both the client and server, a removeClient() method for the server, and a List/ArrayList in the server application to store each created client object. When a client connects to the server the client object is always added as the last element into the ArralistList, and the current index of said object in the ArrayList is passed as a parameter to the client object to represent the client’s number. When a client wishes to exit the application, either by typing “terminate” or closing the window, a message is sent to the server to indicate this and then an openConnection flag is set to false, which, causes the client application to close its socket and all open streams.

Because each client knows its own client number, when the server detects said “quit” message from a client, it sets the Boolean variable openConnection to false. This causes the client object’s processConnection() method loop to exit, which then calls method closeConnection(), method removeClient(),passing the client number in the process, and interrupts the client object’s thread. The removeClient() method removes the client from the ArrayList and, if necessary, iterates through all remaining connected clients and decrements their client numbers. Two selection statements are utilized in this method to ensure that the renumbering of clients does not occur when the only remaining client was removed from the ArrayList or if the removed client was the last item in the ArrayList. The latter clause is simply because deletion of the last element doesn’t affect the inserting and/or numbering of a new client.