COMP416: Computer Networks
Project 1 - Part 1
Client-Server Chat App via TCP Sockets

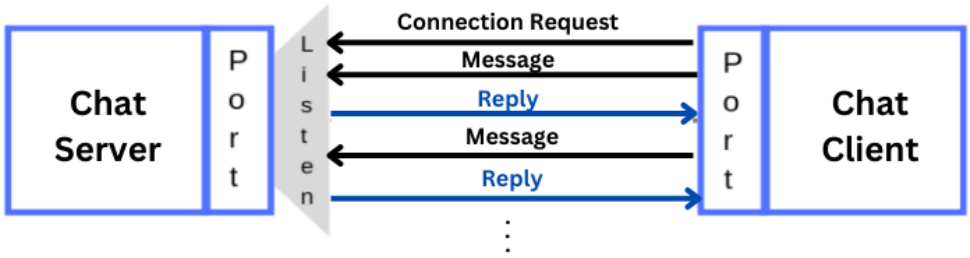
Due: October 21, 8:00pm (Late submissions will not be accepted).

Submission of the project deliverables is via Blackboard.

This is an individual project. You are not allowed to share your codes with each other.

The first part of Project-1 provides relevant hands-on experience with TCP socket programming. It includes programming a Chat Server and a Chat Client, which can communicate with each other on a local machine through a TCP socket. Through this work, you will familiarize yourself with the Java 'Socket' class and some of the methods essential to socket programming.

Sockets:
Sockets provide a means of communication between two machines. Each socket comprises the host address and port. For this project, both the client and the server will be executed on the same machine.



(Successful attempts to execute the chat client on a machine and the chat server on a different machine will result in bonus points awarded).

Project Objectives

You are required to develop a chat application between a server and the client. It is a simple version of a chat application because the client has to start the chat, then the server can reply, and so on.

You can refer to the documentation here for more information on the methods associated with the Socket Class.

The following are the main requirements for this project:

1. Code a chat client and server-side capable of sending each other messages through TCP sockets.
2. The port number at the server and client should not be hard-coded; instead, it should be capable of receiving the port as an argument at the time of execution.
3. Once the connection has been established, both sides should display the complete socket address of the other.
4. Both the client and the server should keep the communication running/ socket open until one of the ends closes the connection through a keyword, e.g., 'goodby, 'see you later'' etc.
5. A timeout should also be enabled for the server socket. This implies that the server should automatically close the socket if no communication takes place between itself and the client. The duration of the timeout should be supplied to the server as an argument at the time of execution.

Project deliverables:

You should submit your source code and project report (in a single .rar or .zip file) via Blackboard.

Report

The report should at max be 3 pages long and should start with a concise ReadMe section to provide instructions for the execution of your code. Following this, you should provide an overview of the programming and attach the snapshots of the execution results depicting each of the project objectives. The report must include the snapshots for the following:

1. The advertised port by the server being provided as an argument at execution
2. Both sides showing the socket address of the other side once the connection has been established.
3. The timeout settings of the connected socket for both sides
4. Chat between the server and client.
5. The keyword being used and chat being ended.
6. The server closes the connection in response to a timeout.

Source Code: A .zip or .rar file that contains your implementation in a single Eclipse or Intellij IDEA project or otherwise.

The report is an important part of your project presentation, which should be submitted as both a .pdf and Word file. Your report should show the step-by-step server-client communication initiation till termination through snippets. The report acts as proof of work for you to assert your contributions to this project. Everyone who reads your report should reproduce the parts we asked to document without hard effort and any other external resource. For codes, you should be taking screenshots instead of copy/pasting the direct code. Strictly avoid replicating the code as a whole in the report or leaving code unexplained.

Good Luck!