CSCI 3171 NETWORK COMPUTING
Assignment No. 4
Date Given: Tuesday, November 19, 2019
Due: Saturday, November 30, 2019, 11.55 p.m. (5 minutes to midnight)
Submission: On dal.ca/brightspace

This assignment focuses on developing multi-threaded server applications using Java socket programming. All the questions are simple extensions of Exercises 3 and 4 in Lab No. 7.
For all the questions, you may assume that one server connects with at least five clients (as in Lab No. 7 exercises), namely, Hans Solo, Luke Skywalker, Darth Vader, ChewBacca and BB-8.

Question No. 1: The objective of this question is to implement a **Chat Server** application. A Chat Server connects with multiple clients and **broadcasts the messages** received by a client to all the **other clients**.

When a client joins, it is prompted with "Enter name: ". The name of the client is read from the keyboard and sent to the server.
The server must recognize that this is the first message and record this name of the client.
Then for each subsequent message, the client is prompted with "Enter message (BYE to exit): "
When the client enters a subsequent message, the server should retrieve the appropriate name and display
Message from <*name of the client*> : *message*
This message should then be broadcast to all the other clients except the one that it is received from. It should be displayed as
Message from <*name of the client*> : *message*
on the console of the other clients.
Note: The message must not be sent back to the client that sent this message.

Question No. 2: In this question, you will extend the program that you wrote for Question 1 **by encrypting the messages using Caesar cipher.**
The server, when started, generates a random key (an integer between 1 and 25).
When a client joins, it is prompted with "Enter name: ". The name of the client is read from the keyboard and sent to the server.
The server must recognize that this is the first message and record this name of the client.
The server then sends the secret key to the client.
The same key is sent to every client that joins the session.
Following this, every message from every client gets encrypted and sent to the server. The server displays the encrypted message and broadcasts the encrypted message to the other clients. The other clients must decrypt the message and display it.

Question No. 3: In this question, you will further extend the program that you wrote for Question 2. **Now each pair of clients must get a separate key from the server so that they can communicate with each other using encrypted messages.**
When a client (say Hans Solo) joins, it is prompted with "Enter name: ". The name Hans Solo is read from the keyboard and sent to the server.
Next, the client is prompted with "Who do you want to communicate with?: ". The name of the client that it wants to communicate with (say ChewBacca) is read from the keyboard and sent to the server.
If the client ChewBacca has not joined the session yet, or if Hans Solo enters a name that is not on the list, then the server must respond with "No such client".
If ChewBacca has already joined the session, the server should generate a reandom secret key (between 1 and 25) and send this key to both Hans Solo and ChewBacca.

From now on, every message between Hans Solo and ChewBacca must be encrypted with the secret key using Caesar cipher.
Note that there is no broadcast in this exercise. It is just a one-to-one communication.

Note also that multiple pairs of clients such as {Hans Solo, ChewBacca}, {Darth Vader, Luke Skywalker}, {BB-8, ChewBacca} can be carrying on conversations at the same time using different secret keys.

Have fun!

Submission instructions: A zip file containing original source codes (not PDFs) and sample screenshot outputs.