Lab 1: Socket Communication

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket serves as a server, listening on a particular IP and port, while another socket serving as a client, establishes connection to the server to communicate with it.

Task

This is an **individual** work. Create a simple chat program, which contains two sockets, one is server and the other is client. Establish connection between two sockets and then exchange messages.

Steps

1. Create a socket as Server ***S***, constantly listening on a specific IP address and port.
2. Create another socket as Client ***C***, and connect to the Server ***S***.
3. Once the connection established, the Client ***C*** first sends a message to the Server ***S***, e.g., *"Hi server, this is client."*
4. The Server ***S*** receives the message from ***C***, print it out and then sends a message back to the Client ***C***, e.g., *"Hello, nice to meet you."*
5. The Client ***C*** receives the response from ***S***, and print it out.

Languages and Libraries

No restriction. Pick up whatever you are familiar with.

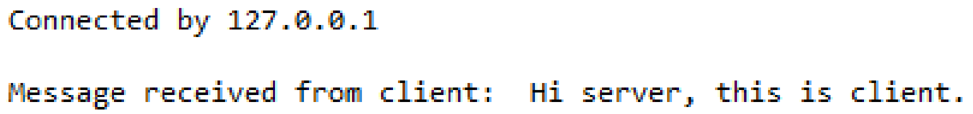
Submission

Submit the following files in a **zip** onto Canvas.

1. All of your source code files for this lab. Code comments will be helpful for the TA to understand your code.
2. A README file including information:
   a) Which language and external libraries you use.
   b) Which IDE you use.
   c) Detailed steps about how to run your code to finally obtain the required outputs. TA will check and run your code based on this instruction. If your code cannot be successfully executed, points will be deducted. **This is the most important part used to evaluate your work, so please be as specific as you can.**
3. A lab report including screenshots of each step of the entire code execution procedure, **as clear and specific as you can**. Alternatively, you can choose to use screen recording to record the entire code execution procedure and submit the recorded file.

Screenshots example

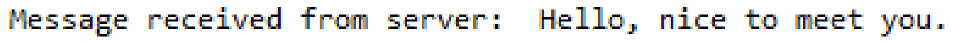
Server side display:



```
Connected by 127.0.0.1

Message received from client:  Hi server, this is client.
```

Client side display:



```
Message received from server:  Hello, nice to meet you.
```

Submission Due: 11am, February 2, 2021.