CN Lab (MA3105) Assignment 1: Socket Programming

Objective

Implement a TCP-based **client-server** application where the client sends its name and a number to the server, and the server responds with its own name and another number. Both client and server will display received values and compute their sum.

Client Requirements

1. Accept an integer between **1 and 100** from the keyboard.
2. Open a TCP socket connection to the server.
3. Send a message containing:
   - A string with your name (e.g., `"Client of John Q. Smith"`).
   - The entered integer value.
4. Wait for a reply from the server.
5. On receiving the reply:
   - Display:
     - Client's name
     - Server's name
     - Client's integer
     - Server's integer
     - The sum of both integers
6. Terminate after releasing all sockets.

Server Requirements

1. Create a string with your name (e.g., `"Server of John Q. Smith"`).
2. Listen for incoming TCP connections from clients.
3. For each received client message:
   - Extract and display:
     - Client's name
     - Server's name
   - Pick an integer between **1 and 100** (can be the same for every client).
   - Display:
     - Client's integer
     - Server's integer
     - The sum
   - Send back:

  - Server's name
  - Server's integer value

4. If the server receives a number outside **1–100**, close all sockets and terminate.

Interaction with Classmates (Interoperability Test)

- Team up with a classmate and test your client with their server, or vice versa.
- Record and submit the output from one side (client or server) showing:
  - Both names
  - Numbers exchanged
  - Correct sums

Programming Notes

- Use a server port number **greater than 5000**.
- Ensure all sockets are closed after use to avoid port binding errors.
- The client and server can run on the same machine or different machines without modification.
- If needed, find your IP address using:
  - **Linux/Mac**: `ifconfig`
  - **Windows**: `ipconfig`
- Optional: Implement a **concurrent server** using threads (Python: `threading`, Java: `Thread`, C: `fork()`).

Submission Method

You may submit your work using **either** of the following methods:

Option 1 — GitHub

1. Create a **public GitHub repository**.
2. Upload:
   - Client and server source code
   - Screenshots/output logs
3. Share the **repository link**.

Option 2 — Google Drive

1. Create a **folder** in Google Drive.
2. Upload:

- Client and server source code
- Screenshots/output logs

3. Set sharing permissions to "Anyone with the link can view".
4. Share the **Google Drive link**.
