CIS656 – Distributed Systems – Fall 2025
Programming Assignment 2 – Multi-Thread Socket Programming
Maximum Points: 100 (3% of the final grade)
Due: 8:30PM on Wednesday, September 17, 2025
(Please finish this programming assignment individually, or as a group of two or three people)

Objective

The objective of this programming assignment is for students to apply their knowledge of multi-threads to network programming. Upon completion, students will be able to apply multithreaded programming techniques to create efficient server solutions in building distributed applications.

Description

In this programming assignment, a client-server communication model is implemented. Different from the first programming assignment, your server should be able to support multiple clients' connections. You need to create two programs, server and client. They should work as follows.

Run the server first. The server should print a statement "The server is running." and waits for the clients' connections.

After the server is up, run the client. Your client should ask your input for the server's IP address.

After the client is connected to the server, the client program shows a welcoming message from the server with the client number. For example, if this client is the first one connected to the server, it shows "Hello, you are client #1". The second client connected to the server therefore shows "Hello, you are client #2", and so on. Then the client asks the user to enter a string to send to the server.

- If a string "time" is entered, the server will send the current date and time to the client and the client prints that.
- If another string other than "time" is entered, then the server will send the capitalized version of the string the user just entered on the client side, to the client for printing.
- Your client program should keep asking the user's input for a string unless an empty string is entered (i.e., just press ENTER key without anything else). When an empty string is entered, this client will close the connection with the server.

Different from programming assignment 1, while a client is being connected to the server, the server should be able to receive new clients' connections for current date/time or capitalized strings. In other words, the server should concurrently handle multiple clients' connections and requests well. And of course, after a client is closed, the server should keep running and be able to receive other clients' connections, unless you manually shut down the server program.

Important Points

- You can open a terminal in the machine running the server program and use the command “hostname -I” to get the IP address of the server. (Here, it is “-I” instead of “-i”.) You will find the IP address in the format as “35.39.29.xx 172.17.0.1”. Please just use the first part “35.39.29.xx” as the IP address for our programming assignments.
- You can choose whatever the programming language you feel comfortable, as long as you use sockets to communicate, multi-threads to support concurrency and have the expected result.
- You should make sure your server and client can work well on different machines. Please use CIS virtual lab machines to program and test.
- Please keep your source files for future programming assignments.

Grading Criteria:

- **Program correctness: 75%**
- **Demo or video: 25% (please see the requirements below. If you only submit the source codes but do not demo your program or make a video, 25% of your grade will be deducted.)**

Depending on the format, please follow the following requirements.

(1) If you can demonstrate your program in person with the instructor, please do that during the office hours, or at the end of Wednesday, September 17’s class. Then you do NOT need to submit anything to BlackBoard.

(2) If you want to demonstrate your program online synchronously with the instructor, you can do that during the office hours, or at the end of Wednesday, September 17’s class online via Zoom. Then you do NOT need to submit anything to BlackBoard.

(3) If you cannot demonstrate your program in person or online synchronously with the instructor, please make a video to show the program correctness and go through your source codes to explain how they work. Then please submit the source codes and video link to BlackBoard. If you work in a group, only ONE group member needs to submit the source file and the video link, but please include your group members' names in "Submission" field of the submission page.