CIS656 – Distributed Systems – Fall 2025
Programming Assignment 3 – RPC
Maximum Points: 100 (3% of the final grade)
Due: 8:30PM on Wednesday, October 1, 2025
(Please finish this individually, or as group of two or three people)

Objective

The objective of this programming assignment is for students to get familiar with RPC and understand how to write programs in Client-Server model using RPC. Upon completion, students will be able to use RPC to create communication between the server and clients.

Description

Please download "RPC-Client.zip" and "RPC-Server.zip" from BlackBoard. They provide RPC examples to perform a simple addition method initiated by the client.

First, please study the sample code and understand how RPC works in this simple remote addition call.

Next, follow the logic of RPC in the example to modify the code. After modification, your programs should work as follows.

- Run the server first. The server should print a statement "The server is running." and waits for the clients' connections.
- The client asks the user to enter a string. (Instead of providing two integers as arguments for RPC, the client calls a remote method in the server using the string from the keyboard input as the argument.)
    - 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.

Finally, further modify your code to enable the server to support connections from multiple clients. 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.

You must use RPC to exchange messages between the server and the clients. You should make sure your server and clients can work well on different machines. Please use CIS virtual lab machines to program and test.

You are recommended to use Java to finish this programming assignment, since the given example is written in Java. However, if you prefer Python, you may take a look at the example in the following URL, modify the code and finish the tasks based on that. https://docs.python.org/3/library/xmlrpc.server.html
More info can be found at https://docs.python.org/3/library/xmlrpc.client.html

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, October 1'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, October 1'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.