

## Part 1 Design Note

The code in part1 implements a simple stock lookup server and client using socket programming in Python. The server maintains a catalog of stocks along with their price and volume. The client can connect to the server and send a request to lookup a stock's price. The server processes the request and responds with the price if the stock is found and has volume remaining, else it responds with 0 if the stock is found but has no volume remaining or -1 if the stock is not found.

Design choices:

1. **Multithreading:** The server code implements a thread pool to handle multiple client connections simultaneously. The thread pool maintains a work queue of incoming client requests, and each worker thread picks a request from the queue and processes it. The use of multithreading improves the server's performance by allowing it to handle multiple requests concurrently.
2. **Locking:** Since multiple threads are accessing the shared work queue, a threading lock is used to ensure thread safety. The lock is acquired before accessing the queue and released after the operation is complete.
3. **Catalog data structure:** The stocks and their details are stored in a Python dictionary. This data structure allows efficient lookup of stocks by name, and their price and volume can be easily updated.
4. **Request and response format:** The client sends the stock name as a string encoded in utf-8. The server processes the request and sends the price as a string encoded in utf-8. The client decodes the response to get the price. If the stock is not found, the server sends -1 as a string encoded in utf-8.
5. **Latency measurement:** The client code measures the latency of each request by calculating the time taken between sending the request and receiving the response. The latency for each run is added, and the total latency for all runs is calculated.
6. **Testing:** The client code uses a list of stock names to test the server's functionality. The list contains valid and invalid stock names to test the handling of different types of requests. The client code also prints the server's response for each request to validate the correctness of the server's response.

Overall, the design choices ensure that the stock lookup server and client are efficient, scalable, and reliable. The code can be easily modified to add new functionality or adapt to different use cases.