Weekly Assignment – Session 6 – 35 pts

Goal: The goal of this project is to build an exchange for two traders.

Special Note

1. This assignment focuses on client/server and numpy

Description

A trader (Sir Liquidity from London) has a lot of liquidities and wants to get rid of them.
Another trader (Mr Long from Chicago) wants to buy only when the moving average of the last 10 values have been increasing 3 times in a row.
Since they don't want to trade directly because they prefer staying anonymous, they need to use an exchange. This exchange will be located in New York.
This exchange is very simple and is not a real one. It will be receiving the price given by Sir Liquidity and will transfer this price to Mr Long.
Mr Long will calculate the moving average for the last 10 received values. If this moving average increases three times, Mr Long believes that the price will increase again and will send an order to the exchange.
The exchange only accepts "BUY" and "SELL" from Mr Long and the prices from Mr Liquidity.
The exchange only sends prices to Mr Long and sends "BUY" and "SELL" to Mr Liquidity.
Mr Liquidity will send price following the uniform distribution defined by this numpy vector.

```
>>> np.random.seed(12345)
>>> sampl = np.random.uniform(low=12.5, high=14.5, size=(10,))
```

Please don't forget the seed(12345) otherwise your results will be different.
You should have the first 10 elements of your vector as the following:

```
>>> print sampl
[ 14.35923219  13.13275111  12.86783762  12.90912056  13.63545006
  13.69108941  14.42902904  13.80635419  13.99781328
13.80713974]
```

These two parts will be running using different code.
So you need to have 3 different files (exchange.py, long.py, liquidity.py)

Part 1 (10 pts) – Exchange (exchange.py)

Create a class Exchange accepting two connections (Mr Long and Sir Liquidity).

This class is a TCP server accepting two connections.

This exchange will:

- Accept two connections on the port 9999 (Mr Long) and 9998 (Sir Liquidity)
- Receive prices from Sir Liquidity (there is no side, no quantity)

- Send prices to Mr Long (there is no side, no quantity)
- Receive “SELL” or “BUY” from Mr Long
- Send “SELL” or “BUY” to Sir Liquidity

Part 2 (10 pts) – Mr Long (long.py)

Create a class Long.

This class:

- Connect to the server Exchange using the port 9999
- Receive prices from the Exchange
- Calculate moving average for the last 10 points
- When the moving average increases 3 times, sends “BUY” to the exchange for the price trigging this order.
- When a BUY has been sent, Mr Long wait for a drop in the price and sends “SELL” to the exchange
- When Mr Long already sends a “BUY” he cannot sends another “BUY”, the only order he can send is “SELL”
- When Mr Long sends a sell he will calculate the gain or the loss and will accumulate that in the variable “balance”

Part 3 (10 pts) – Sir Liquidity (liquidity.py)

Create a class Liquidity.

This class:

- Connect to the server Exchange using the port 9998
- Send prices reading the vector of prices generated using numpy. The size of this vector should be 500. The speed to send a price will be every second.
- When a “BUY” has been received, Mr Long records the price and when a “SELL” is received, he will calculate how much money he won or loss and it will store his balance in the variable “balance”

Part 4 (5 pts) – Verification

When Sir Liquidity and Mr Long are done trading (after 500 prices), they display their Profit and Loss (P&L). The absolute value of balance for these two people must be the same (of course one of them will lose money and the other will make money).

The exchange doesn’t need to keep track of the P&L.