

JPM Quant Mentorship 2024

Optional Assignment for Programming

Work in teams of 2 or 3 to implement the below.

We recommend using OOPs concepts to implement the below using good programming practices.

In this exercise, you will simulate a stock exchange and traders interacting with it.

The major components are described below, feel free to create more entities as required.

1. A Stock Exchange:

Which constitutes

- a. A method that indicates the last traded price of any security
- b. A method that indicates the current best bid and best offer
- c. A method that indicates the top 5 bids and top 5 offers.
- d. A method to accept bids or offers
- e. Order Matching Engine (
 - i. The Matching engine must always follow price-time priority for all orders
Price time priority: https://www.marketswiki.com/wiki/Price-time_priority
 - ii. When a trade is matched, the exchange should collect money from the person buying the share and send it to the person selling it. The share should also change hands immediately.
 - iii. Assume the exchange does not charge any fees for facilitating trades.
 - iv. The Engine should only accept orders sent during the trading hours.
 - v. Any bid or offer outside of the top 5 bids or top 5 offers is cancelled and the trader is notified. Any pending orders at the end of the day are cancelled and the trader is notified.

2. An Order management system for a trader

- a. A method to track the trader's cash in the trading account
- b. A method to add and withdraw cash from the trading account.
- c. A method that indicates the current value of the trader's portfolio
- d. A method to place buy orders.
- e. A method to place sell orders.

3. Trader

- a. Has access to an Order management system
- b. Has a bank account with a fixed amount of money in it.
- c. An action
 - i. An action is defined as either a buy or a sell decision on a stock
 - ii. The price at which the buy or sell order is placed should be randomized between the below three options
 1. Place order at best bid
 2. Place order at best ask
 3. Place order at mid price ($\text{mid price} = (\text{best bid} + \text{best ask}) / 2$)
 4. Assume the size of the order placed (quantity) is always 1000.

5. If there are no bids or offers present in the market, the trader places a buy or sell order arbitrarily 5% above or below the previous closing price.

4. Share

- a. All shares are identical to each other.

The stock exchange matching engine must take orders as they come and process them as quickly as possible while maintaining price time priority. Assume there is only one exchange possible. The stock exchange must support order matching on up to five securities.

Any new order that arrives will be queued in the engine.

An order can be placed only via the order management system (OMS). The OMS must send orders to the exchange and the Exchange will notify the OMS when an order has been fulfilled.

Details surrounding the security and relationship between objects has been intentionally left vague, feel free to use your best judgement when the relationship is unclear.

Simulation:

Create 5 Trader instances with an arbitrary amount of money in their bank accounts and a random collection of stocks in their portfolio. i.e. they start with a non-zero portfolio.

Assume time moves in 1 second steps. At each second, the traders should arbitrarily perform a buy action or sell action. The price to buy or sell at should be decided randomly as described in the prior section.

If a trader runs out of money during the trading day, they can deposit more money into their trading account if they want to. If they run out of money or stocks to sell, they stop taking any actions.

Simulate this scenario for a day assuming an 6.5 hour trading day and display the profits (or loss) of each trader at the end of the day.

Submission format:

Please email us your code and screen shots of the output with reasonable logging to indicate what is going on. Zip everything together into a single file and send it over to us, along with details on the team members. Please do not share jupyter notebooks with us.

For any clarifications, please email: jpmqrmentorship.mumbai@jpmorgan.com and cc me: sai.kumarm@jpmchase.com .