



## 1 A mini brokerage system

In this problem, the objective is to create a toy **brokerage firm software system** while implementing OOP techniques. We will need to know some pre-requisites as below.

- A **publicly listed company** is a company whose ownership is distributed among general shareholders through a *stock exchange*. For example, Tata Consultancy Services, Google Inc, State Bank of India etc.
- A **stock exchange** or *securities exchange* is a facility where traders can buy/sell shares of publicly listed companies and other securities(i.e., bonds, rights, debentures etc.) mostly electronically. For example New York Stock Exchange, Bombay Stock Exchange etc,
- A person or entity who buys/sells *securities* is called a **trader** or *investor* depending on the context. It could be an individual or a society or an organisation.
- A **brokerage firm** is a liaison between the trader/investor and the stock exchange. For example, Interactive Borkers, Zerodha etc.
- Each listed company in the stock exchanges has a symbol referred to as **ticker**. For example, the ticker for Infosys is *Infy*, the ticker for Mahindra & Mahindra Limited is *M&M*.

Business takes place in Stock exchanges generally on weekdays for a period of 7 to 8 hours. For example, in India there are two major stock exchanges namely NSE and BSE where trading hours is from 9:15 AM to 3:30 PM. During the market hours, the prices of the stocks(i.e., TCS, Hindustan Uniliver, TVS Motors, SBI, DLF etc) keep changing based on the market conditions.

- **LTP** is the latest trading price of a stock at any given time.
- On a given trading day, the price at which the first transaction takes place is called the **open price** and the price at which the last transaction takes place is called the **close price**.
- On a given trading day, for a given stock, the highest price at which a transaction takes place is called the **high price** and the lowest price at which a transaction takes place is called the **low price**
- The lowest price at which a seller is willing to sell a particular stock is called the **ask price**.
- The highest price at which a buyer is willing to buy a particular stock is called the **bid price**.
- When a trader places an order with the brokerage firm the order is said to be **open**. Then the order(say, buy order) gets matched with another order of opposite type(say, sell order), the order gets executed and is said to be **filled**.
- The **orderbook** is the recordbook of all open buy and sell orders at any given time.
- It is possible to sell the shares of a stock even before owning it. Say you don't own a share of TCS but you think that the price of TCS will fall in the next few hours. In that case you may sell a share of TCS and buy it later during the day. Such type of selling is called **short selling**.

In the figure 1, the bid price is 100.25 where there are orders to buy 100 shares at the bid price. Also the ask price is 101 where there are orders to sell 200 shares at the ask price. Also there are other orders that are kept sorted. A transaction takes place when the ask and bid prices agree. For example, in the order book 1 if there is a new sell order placed to sell 50 shares at INR 100.25, then a transaction will take place. The brokerage system has to allow the following essentially.

Buy Orders		Sell Orders	
Shares	Price	Shares	Price
100	100.25	200	101
300	100	300	101.25
500	99	600	102

Figure 1: A sample order book

- You should implement two **stock exchanges** namely NSE and BSE. A given stock may be registered to either or both of these exchanges.
- **Add** new stocks into the system for trading. Suppose a new company is listed in the stock exchange, then in your program, you have to implement that by adding a new instance of your relevant class. The company should have a unique identifier or **ticker**. It should be possible to query *Open*, *High*, *Low*, *Close* values for a given stock.
- **Delete** an existing stock from the system if the company is delisted from the stock exchange. Such things happen in the stock market.
- **Add** new trader into the system. When a new trader registers himself with the broker you have to add her into the system by instantiating appropriate classes in your program. The trader should have a unique *customer ID*. It should be possible to query the list of shares of stocks she owns. It should be possible to query the set of open orders she has placed.
- Enable placing **Buy/Sell** orders. Also ensure that the buyer has sufficient funds when she places a order. You should implement appropriate methods in your program.
- **Execute** transactions when *Ask* and *Bid* prices agree. You should implement appropriate methods in your program.

[6 points]

Now there is the requirement to classify the stocks into types such as Pharma, Consumer Goods, Automobiles, IT, Infrastructure, Finance, Banking etc. So you should be able to have appropriate attributes and methods in your program to implement that. It should be possible to query the set of Pharma companies on a given stock exchange.

[1 point]

**Definition 1.** For a given stock, the maximum price at which a transaction can possibly take place on a given day is called the **upper circuit** and the minimum price at which a transaction can possibly take place on a given day is called the **lower circuit**. Exchanges decide upper and lower circuit each trading day. It is generally Last Close Price  $\pm 5\%$  or Last Close Price  $\pm 10\%$ . For the purpose of your program keep upper circuit at Last Close Price  $+10\%$  lower circuit at Last Close Price  $-10\%$ .

If the user places a order which is below the lower circuit for a stock, the system should raise an invalid input exception with appropriate message. Similar thing should happen for upper circuit violation. [1 point]

You are expected to think through the above scenarios and about how the above may be implemented. Try drawing a class diagram and analyze what attributes and methods each class should have. Then think about the possible interactions between the classes.

[2 points]

Now at the end of the day the whole business of participating in the stock market is to make money. So let us come to the more interesting part.

**Definition 2.** *P/L* which stands for Profit/Loss is *selling price - buying price* for a given stock. If you bought 10 shares of TCS at INR 2100 each and sold them at 2500 each then your  $P/L = 10 \times (2500 - 2100) = 4000$ .

**Definition 3.** The *return* of a stock is the profit/loss made by buying and selling of that share over a period of time.

**Definition 4.** The maximum loss a person can make by buying and selling once during a given period of time is called the max drawdown. Formally, the difference between the highest price and subsequent lowest price for a given stock during a given time is called the **Max Drawdown**. Note that it is not simply same as the difference between the highest and lowest price during a given time.

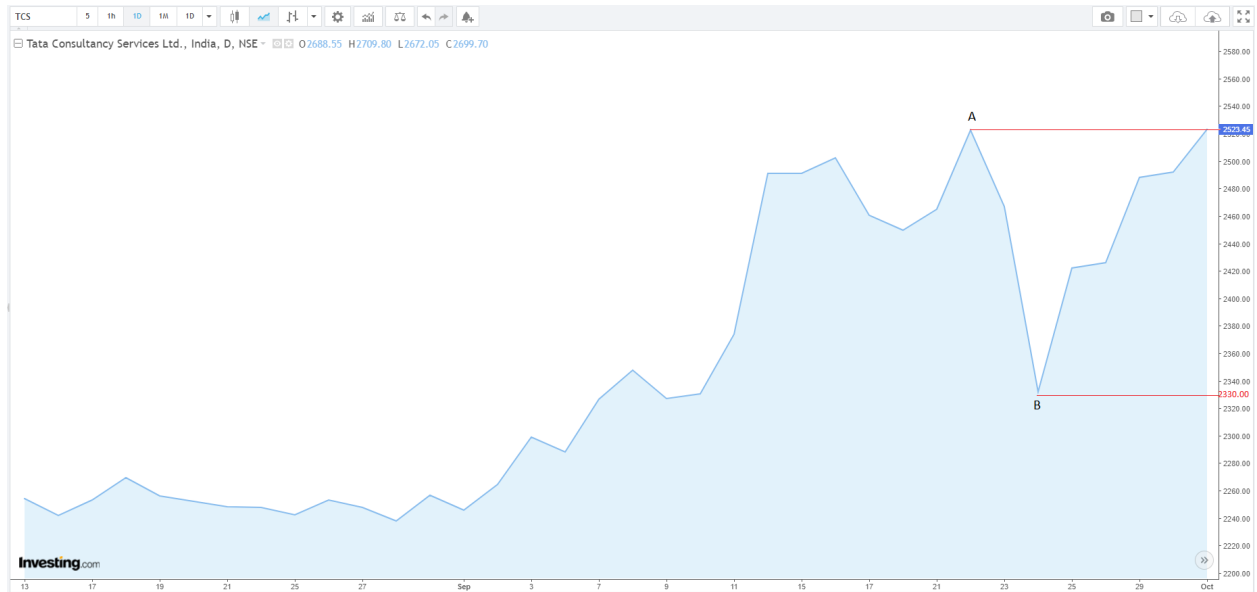


Figure 2: A sample stock price chart showing TCS price from 13th August to 30th September, 2020. Courtesy: Investing.com.

Please have a look at the price chart of TCS in chart 2. The x-axis is time and y-axis is price. Observe that if somebody were unlucky enough to buy at the peak at point A and sell at the trough at point B, she will experience the maximum loss over the given time period. This loss, in this case  $2523.45 - 2330 = 193.45$  per share, is the max drawdown.

**Definition 5.** Let us define the **max return potential** of a given stock for a given period of 15 trading days. If the *close price* is more than the *open price* on a given day, we say that the stock is bullish on that day. Similarly if the *close price* is less than the *open price* on a given day, we say that the stock is bearish on that day. Now suppose you were lucky enough to buy at the open price and sell at the close price on each bullish day and sell at the open price and buy at the close price on each bearish day for a period of 15 consecutive trading days. Your net profit at the end of the 15 trading days will be termed as **max return potential** of that stock. We compute the *max percentage return potential* as  $100 \times \text{max return potential} / \text{open price on day 1}$ .

Now you are expected to implement some additional features to meet the following use-cases.

- Given data for a period of 15 trading days, calculate the average price of a given stock.
- Given data for a period of 15 trading days, calculate the *max drawdown* of a stock. While calculating *max drawdown* consider the daily close prices into consideration.
- Given data for a period of 15 trading days, calculate the *max return potential* of a stock. While calculating *max return potential*, please ensure to report the *max percentage return potential*. Refer to the definition 5. [2 points]

Tip: You may maintain a list of hashmaps to keep the last 15 days O, H, L, C data of a given stock.

## 2 Input

Two input text files will be given. You should be able to process the input by doing File I/O operations. Find below two sample input files.

————input.txt————

Add scrip: INFY, sector: IT, O:890, H:890, L:890, C:890

Add scrip: TCS, sector: IT, O:2250, H:2250, L:2250, C:2250

Add scrip: SBI, sector: Bank, O:190, H:190, L:190, C:190

Add scrip: M&M, sector: Auto, O:610, H:610, L:610, C:610

Add scrip: Cipla, sector: Pharma, O:790, H:790, L:790, C:790

Add scrip: Sunpharma, sector: Pharma, O:490, H:490, L:490, C:490

Add user: Jaydeep, funds:15000 holding: None

Add user: Mimi, funds:1000 holding: {INFY:10, TCS:5, SBI:20}

Add user: Kapil, funds:25000 holding: {SBI:100, M&M:20}

Add user: Nusrat, funds:2000 holding: {INFY:20, M&M:25, SBI:25}

Place order, user: Jaydeep, type: buy, scrip: INFY, qty:10, rate: 790

Place order, user: Jaydeep, type: buy, scrip: SBI, qty:100, rate: 210

Place order, user: Mimi, type: sell, scrip: INFY, qty:10, rate: 900

Place order, user: Kapil, type: buy, scrip: M&M, qty:10, rate: 580

Place order, user: Nusrat, type: sell, scrip: SBI, qty:25, rate: 195

Place order, user: Mimi, type: buy, scrip: TCS, qty:5, rate: 2190

Show Orderbook

Execute

Show sector: Pharma

Delete scrip: TCS

Delete User: Jaydeep

Delete scrip: M&M

Delete User: Kapil

Show Scrips

Show Users

Exit

---

Historical data of a period of 15 days will be given in a comma separated values format as below.

—INFY\_15days\_data.csv—

Ticker,Date,Prev Close,Open Price,High Price,Low Price,Last Price,Close Price

INFY,4-May-20,715.5,689.8,701.2,670.35,675.15,673.7

INFY,5-May-20,673.7,681.75,685.6,670,674.5,673.7

INFY,6-May-20,673.7,676.9,681.45,664.05,667,665.9

INFY,7-May-20,665.9,665.55,674.35,662.45,666.7,664.95

INFY,8-May-20,664.95,672.25,680.9,668.5,676,674.2

INFY,11-May-20,674.2,682.5,698.55,678,683.75,682.9

INFY,12-May-20,682.9,678.5,691,675.5,687,687.65

INFY,13-May-20,687.65,694.8,699.9,683.75,693.75,694.15

INFY,14-May-20,694.15,659,671.75,652.6,658.2,658.15

INFY,15-May-20,658.15,664.8,664.8,646.7,652.3,652.3

INFY,18-May-20,652.3,659.85,670.55,653,659.6,663.9

INFY,19-May-20,663.9,664,674.8,655,670,668.6

INFY,20-May-20,668.6,668.5,673,659.65,666,669.55

INFY,21-May-20,669.55,666.5,679.45,665.4,673,672.2

INFY,22-May-20,672.2,675,695.85,673.5,692,692.35

---

Tip: Open the sample file INFY\_15days\_data.csv in MS Excel for ease of viewing.