

Programming Project: Stock Exchange

A stock exchange is an organization for trading shares in publicly owned companies. In the OTC (“Over the Counter”) system, stocks are traded electronically through a vast network of dealers in securities connected to a computer network. There is no physical “stock exchange” location. The system keeps track of “buy” and “sell” orders placed by customers through their brokers and automatically executes orders when the highest “bid price” (order to buy shares at a certain price) meets the lowest “ask price” (offer to sell shares for a certain minimum price). There are also “market” orders to buy or sell shares at the current “bid” or “ask” price. With the advent of modern technology, some electronic brokerages let customers place their own orders through their personal computers over a modem or computer network.

Shares are normally sold in units of 100. The stocks are identified by their four- or five-letter trading symbols, all capital letters. For example, Microsoft is “MSFT” and Intel is “INTC.”

Programming Project: Stock Exchange

In this project you will implement a toy OTC system for trading in a dozen stocks. The system will maintain the list of all active buy and sell orders for each stock. In your system a customer can log in, give his or her name, request quotes (current prices) and place orders. An order holds the name of the customer, a buy or sell indicator, the stock symbol, the number of shares, and a price limit (or “market”). Your system will not keep track of the availability of money or shares on “customer” accounts.

Stock prices are traditionally quoted in dollars and fractions of $1/2$, $1/4$, $1/8$, etc. For example, “ $52 \frac{1}{2}$ ” means \$52.50, and “ $38 \frac{7}{8}$ ” means \$38.87. For the sake of simplicity, your system may use dollars and cents.

Programming Project: Stock Exchange

Your system should keep all buy orders for each stock in a priority queue ranked by the bid price, and all sell orders in an inverted priority queue with the smallest “ask” price order on top. The priority queues must be implemented through a heap (implemented as an array). After a new order comes in, the system checks if it can be executed and, if so, executes it and reports to both parties as follows:

Sharon K.:

You sold 2000 of MSFT at \$84.37.

Commission \$19.95.

Total proceeds \$168720.05.

Bill M.:

You bought 2000 of MSFT at \$84.37.

Commission \$19.95.

Total due \$168759.95.

Programming Project: Stock Exchange

The system executes a "market" buy order at the price of the lowest ask and a "market" sell order at the price of the highest bid. Normally they can be executed immediately as they arrive. In the unlikely event that there is only a "market" sell and a "market" buy, execute them at the last sale price.

Your system should keep track of the day's "high" and "low" prices and the last sale price for each stock and be able to quote them to customers on demand. The quote should also contain the current (highest) bid and (lowest) ask and the number of shares in them. For example:

MSFT

Hi:	89.50	Lo:	82.87	Last:	84.37
Bid	82.87	Bid size	2000		
Ask	84.00	Ask size	1000		

Programming Project: Stock Exchange

In your system all orders will be “partial” orders. This means that if an order cannot be executed for the total number of shares requested in it, the maximum possible number of shares will change hands and a partial order for the remaining shares will still be active. For bid and ask quotes and partial orders you will need to add a function that “peeks” at the top of the heap without removing an element from it.

In addition to the source code and a working program, develop a plan for testing your program.