

**MF 703 C++ for Mathematical Finance**  
Spring 2015

**Project #3**

In modern financial markets, market makers (dealers) are agents who stand ready to buy and sell securities. Market makers are remunerated for their services by being able to “buy low and sell high”. Instead of a single price at which any trade can occur, dealers quote two prices – a “bid” (dealer’s purchase, customer’s sale) and an “ask” (dealer’s sale, customer’s purchase). The ask price is higher than the bid price, and the difference between the two is called the spread – the dealer’s source of revenue.

An electronic exchange uses limit orders and employs no designated middlemen. All liquidity comes from customers’ limit orders that are arranged in order books (**essentially two priority queues ordered by price**) as shown in following figure (a) (limit price – number of shares). Priority is given to those participants who placed their order first.

...	} Sell Orders	...	...	
25.56 – 300		25.56 – 300	25.56 – 300	
25.55 – 1000		25.55 – 1000	25.55 – 1000	
25.35 – 200		25.35 – 200	25.35 – 100	
25.30 – 150		25.30 – 150		
<hr/>				
25.21 – 200	} Buy Orders	25.21 – 200	25.21 – 200	
25.19 – 300		25.20 – 1000	25.20 – 1000	
25.15 – 785		25.19 – 300	25.19 – 300	
25.10 – 170		25.15 – 785	25.15 – 785	
...		25.10 – 170	25.10 – 170	
		...	...	
(a)	(b)	(c)		

If a new order arrives, and there are no orders on the opposite side of the market that can satisfy the limit price, then the order is being entered into the book. In (b), a new buy order for 1000 shares at \$25.20 or less has arrived, but the best sell order is for \$25.30 or more; thus no transaction is possible, and the new order is entered into the buy queue. When another buy order arrives for 250 shares at \$25.40 or less, it gets transacted (or crossed) with the outstanding orders in the sell queue: 150 shares are bought at \$25.30 and another 100 shares are bought at \$25.35 (see (c)).

- 1) Write an Order class which incorporates the principal of encapsulation and provides the following members:
  - a. Security: ticker, as string data type
  - b. Side: buy or sell, as string data type
  - c. Type: limit or market, as string data type
  - d. Qty: number of shares, as an integer data type
  - e. Limit Price: price if order type is limit.
  - f. Counterparty Name: as string data type
- 2) Write an OrderBook class which incorporates the ability to perform the following:
  - a. Accepts an order from a market participant. Please send an acknowledgment that the order was received.

- b. Regardless of order type, try to fulfill the order. If not possible, please place the order (if limit) in the order book. Please send a message to the market participant of the status of the order.
  - c. If an order has been fulfilled, whether partially or fully, send a message to the market participant of the order status.
- 3) Write a Message class which incorporates all functionality for a dialogue between exchange and market participant. Please note that not all possible message types were discussed in 2). Think about all possible messages that a market participant will need to know.
- 4) Input will be orders of class type Order can come from market participants. For this assignment, you can assume that input comes from a file of the form (security, side, type, qty, limit price, counterparty name).

ABC, B, limit, 1000, 24.50, Giant Hedge Fund  
ABC, B, limit, 500, 24.75, Namini Fund  
ABC, S, market, 80, 0, BU Investment Fund  
Etc...

I will provide a file if you like, but again, I want your application to as general as possible.

Note: The hard part of this assignment is building the correct data structure of the Orderbook so as to implement the price-time priority.