

## Assignment 3

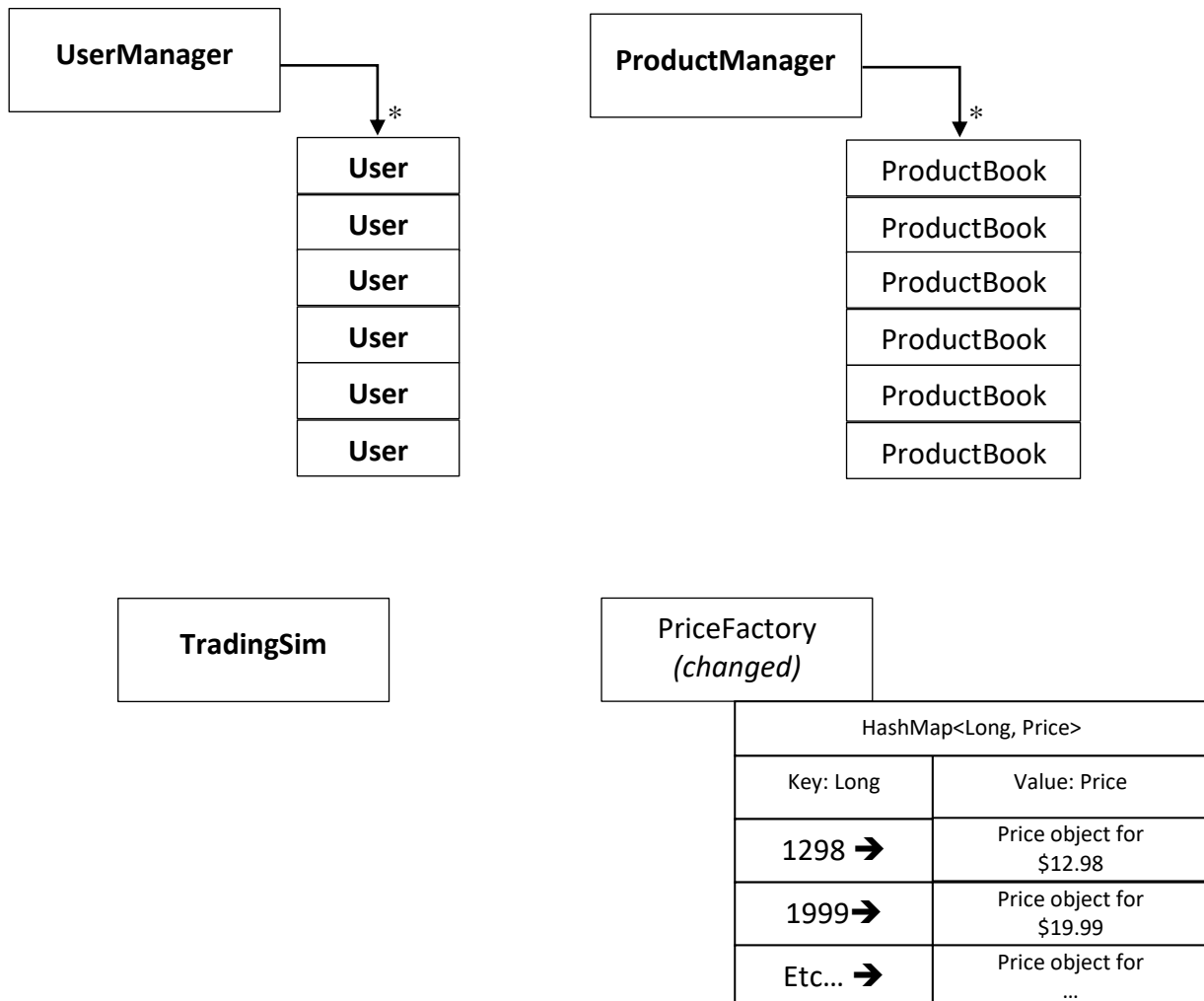
### Object-Oriented Software Development

#### Adding Users, Book Management, and Trading Simulator (200 pts)

#### Objectives

Create well-written object-oriented classes that represent trading Users, a UserManager (a Façade to the Users), a ProductManager (a Façade to the ProductBooks, and a TradingSim class to simulate trading user behavior. The PriceFactory class should be changed to implement the Flyweight design pattern. You should also create any new exception classes if needed to handle invalid situations.

#### Classes and their relationships (new classes in bold)



## Assignment 3 Classes

### 1) User class

The User class represents a trader user. Trader users will enter orders for trading in our system. The user will also maintain a collection of orders submitted, as described below:

- The User class should have a public constructor that accepts and sets the 3-character user id (described below).
- A User must maintain certain data values – shown below (remember to make these private to enforce information hiding):
  - String `userId`: A 3-letter user code – must be 3 letters, no spaces, no numbers, no special characters (i.e., XRF, BBT, KNT, etc.). This should be passed into the constructor. Cannot be changed once set.
  - `HashMap<String, OrderDTO> orders`: The user class should maintain this `HashMap` of orders (DTOs) they have submitted. The key is the order id, the values is an order. This should be set to a new `HashMap` when it is declared.
- A User must be able to perform certain behaviors – shown below. They should be public unless otherwise indicated:
  - Create an accessor for the `userId`.
  - Create a *private* modifier for the `userId` that is called by the User constructor. The modifier should ensure the incoming value meets the field requirements listed above.
  - `public void addOrder(OrderDTO o)` → Add the incoming order DTO to the user's order hashmap. The incoming order should not be null.
  - `boolean hasOrderWithRemainingQty()` → Returns true if any of the order DTOs in the user's order `HashMap` have a `remainingQuantity > 0`. Return false if not.
  - `OrderDTO getOrderWithRemainingQty()` → Return *any* order from the user's order `HashMap` that has a `remainingQuantity > 0`.
  - Override the `toString` method to generate a String like this:
    - User id, then the user's orders.

#### Example

User Id: CAT

Product: GOOG, Price: \$52.45, OriginalVolume: 220, RemainingVolume: 0, CancelledVolume: 0, FilledVolume: 220, User: CAT, Side: SELL, Id: CATGOOG\$52.4557855274865100

Product: GOOG, Price: \$52.40, OriginalVolume: 270, RemainingVolume: 270, CancelledVolume: 0, FilledVolume: 0, User: CAT, Side: SELL, Id: CATGOOG\$52.4057855298770300

Product: GOOG, Price: \$52.85, OriginalVolume: 70, RemainingVolume: 40, CancelledVolume: 0, FilledVolume: 30, User: CAT, Side: SELL, Id: CATGOOG\$52.8557855267845000

## 2) UserManager class

The UserManager class maintains a collection of all users in the system – it acts as a Façade to the Users. Since we only want to have one UserManager object in our application, this class should be a Singleton (design pattern).

- The UserManager maintains a collection (i.e., HashMap) of all users. The key is the userId, the value is the User object.
- The UserManager must be able to perform certain behaviors – shown below. They should be public unless otherwise indicated:
  - `void init(String[] usersIn)` → Create a new User object for each userId in the String array passed in. Each User object should be added to the UserManager's HashMap of users.
  - `User getRandomUser()` → Return a randomly selected User object from the UserManager's collection of users.
  - `void addToUser(String userId, OrderDTO o)` → Add the OrderDTO to the specified User (using the User's "addOrder" method).
  - Override the `toString` method to generate a String like this:
    - The result of calling each User's `toString()`.

Example:

User Id: ANN

Product: GOOG, Price: \$52.95, OriginalVolume: 270, RemainingVolume: 270, CancelledVolume: 0, FilledVolume: 0, User: ANN, Side: SELL, Id: ANNGOOG\$52.9558518804111600

Product: WMT, Price: \$70.40, OriginalVolume: 305, RemainingVolume: 155, CancelledVolume: 0, FilledVolume: 150, User: ANN, Side: BUY, Id: ANNWMT\$70.4058518805034300

...

User Id: BOB

Product: TGT, Price: \$88.75, OriginalVolume: 300, RemainingVolume: 300, CancelledVolume: 0, FilledVolume: 0, User: BOB, Side: BUY, Id: BOBTGT\$88.7558518798475100

Product: WMT, Price: \$70.59, OriginalVolume: 210, RemainingVolume: 210, CancelledVolume: 0, FilledVolume: 0, User: BOB, Side: SELL, Id: BOBWMT\$70.5958518807798900

...

## 3) PriceFactory class - changes

You should update the PriceFactory class to implement the Flyweight design pattern. This means that the PriceFactory should never generate a Price object for the same value more than once. (*Example: 100 requests for a Price object for \$10.00 would generate 1 Price object the first time it was requested, and a reference to that same object the remaining 99 times*).

Your PriceFactory should maintain a HashMap of all Prices created (the key is the Long cents value, the value is the Price object). Every Price object created should be added to the HashMap. Any requests to make a Price object should first check the HashMap to see if a Price object for that value has already been created – if so, return a reference to that existing Price object. If not, create the Price object, add it to the HashMap, then return the Price object.

The Price class should not need any changes when making the above changes.

#### 4) **ProductManager** class

The ProductManager class maintains a collection of ProductBook objects for all stocks used in the system – it acts as a Façade to the ProductBooks. Since we only want to have one ProductManager object in our application, this class should be a Singleton (design pattern).

- The ProductManager will need the following data elements to properly represent a product book in the application:
  - A collection of all ProductBooks in the application. This is best represented as a HashMap (the key is the String product symbol; the value is a ProductBook object).
- The ProductManager will need the following methods to properly manage the ProductBooks in the application:
  - void addProduct(String symbol): This method should create a new ProductBook object for the stock symbol passed in, and add it to the HashMap of all ProductBook objects.
  - String getRandomProduct(): Return a randomly selected product symbol from the collection of all ProductBook objects.
  - OrderDTO addOrder(Order o): Add the Order to the ProductBook (using the String product symbol from the Order object to determine which ProductBook it goes to). Return the OrderDTO you receive back from the ProductBook.
  - OrderDTO cancel(OrderDTO o): Using the String product symbol from the OrderDTO passed in, find the ProductBook. Call that ProductBook's "cancel" method passing it the side and orderId from the OrderDTO passed in. If successful, return the OrderDTO returned from the ProductBook's "cancel" method. If the cancel attempt fails, print a message indicating the failure to cancel, and return a null.
  - public String toString(): Override the toString method to generate a String containing a summary of all ProductBooks as follows (be sure to let the ProductBooks generate their part of the String).

#### Example

```
Product: AMZN
Side: BUY
  Price: $51.15
    DOG order: BUY AMZN at $51.15, Orig Vol: 180, Rem Vol: 135, Fill Vol: 45, CXL Vol: 0,
      ID: DOGAMZN$51.1581541872963500
  Price: $51.00
    CAT order: BUY AMZN at $51.00, Orig Vol: 245, Rem Vol: 175, Fill Vol: 70, CXL Vol: 0,
      ID: CATAMZN$51.0081541863804400
  Price: $50.70
    DOG order: BUY AMZN at $50.70, Orig Vol: 140, Rem Vol: 140, Fill Vol: 0, CXL Vol: 0,
      ID: DOGAMZN$50.7081541863058100
    ANN order: BUY AMZN at $50.70, Orig Vol: 70, Rem Vol: 70, Fill Vol: 0, CXL Vol: 0,
      ID: ANNAMZN$50.7081541866669700
    DOG order: BUY AMZN at $50.70, Orig Vol: 260, Rem Vol: 260, Fill Vol: 0, CXL Vol: 0,
      ID: DOGAMZN$50.7081541872554400
Side: SELL
  Price: $51.30
    DOG order: SELL AMZN at $51.30, Orig Vol: 210, Rem Vol: 210, Fill Vol: 0, CXL Vol: 0,
      ID: DOGAMZN$51.3081541823040600
  Price: $51.35
    BOB order: SELL AMZN at $51.35, Orig Vol: 110, Rem Vol: 110, Fill Vol: 0, CXL Vol: 0,
      ID: BOBAMZN$51.3581541866450900
  Price: $51.50
    CAT order: SELL AMZN at $51.50, Orig Vol: 185, Rem Vol: 185, Fill Vol: 0, CXL Vol: 0,
      ID: CATAMZN$51.5081541846788400
```

## 5) TrafficSim class

The TrafficSim class is a trading behavior driver that creates product books and usrs, then simulates users trading those products (using your classes). This is essentially a test class which will exercise your classes and their methods. The TrafficSim class will be used in a static context – no TrafficSim need be created, the methods and data elements are static. TrafficSim methods can therefore be invoked using the class name, for example: TrafficSim.doStuff().

- In order to trade with realistic prices, the TrafficSim class stores base price values that will be used to generate BUY and SELL prices.
- For test purposes, we will create 5 users and 4 product books.
- The TrafficSim will generate a fixed number of orders (i.e., 100) generated from random users, for random products with generated prices and quantities (volumes).
- The TrafficSim class will need the following static data element to store base prices:
  - Private static HashMap<String, Double> basePrices: This HashMap will be used to associate a product symbol (i.e., WMT) with a base stock price (i.e., 140.98). *Note here we use a Double to store the base price, as it is used for simulation purposes only.*
- The TrafficSim will need the following methods to perform its simulation tasks:
  - public static void runSim(): *See Appendix A: “runSim” method.*
  - private static Price getPrice(String symbol, BookSide side): *See Appendix B: “getPrice” method.*

## 6) Testing

Create a class called “Main” and add to that class a “main” method [public static void main(String[] args)]. In your “main” method, call the TradingSim.runSim() function. *Below is sample output – each run will yield different results.*

- 1) ADD: SELL: CAT order: SELL TGT at \$87.60, Orig Vol: 300, Rem Vol: 300, Fill Vol: 0, CXL Vol: 0, ID: CATTGT\$87.6082076753773700
- 2) ADD: SELL: ANN order: SELL AMZN at \$50.85, Orig Vol: 45, Rem Vol: 45, Fill Vol: 0, CXL Vol: 0, ID: ANNAMZN\$50.8582076805829000
- 3) ADD: SELL: BOB order: SELL TSLA at \$99.15, Orig Vol: 210, Rem Vol: 210, Fill Vol: 0, CXL Vol: 0, ID: BOBTSLA\$99.1582076806203800
- 4) ADD: BUY: EGG order: BUY TSLA at \$97.75, Orig Vol: 240, Rem Vol: 240, Fill Vol: 0, CXL Vol: 0, ID: EGGTSLA\$97.7582076806500700
- 5) ADD: BUY: BOB order: BUY WMT at \$70.65, Orig Vol: 195, Rem Vol: 195, Fill Vol: 0, CXL Vol: 0, ID: BOBWMT\$70.6582076806796500
- 6) ADD: BUY: BOB order: BUY WMT at \$71.15, Orig Vol: 175, Rem Vol: 175, Fill Vol: 0, CXL Vol: 0, ID: BOBWMT\$71.1582076807077000
- 7) ADD: BUY: DOG order: BUY TSLA at \$98.45, Orig Vol: 140, Rem Vol: 140, Fill Vol: 0, CXL Vol: 0, ID: DOGTSLA\$98.4582076807364900
- 8) ADD: BUY: EGG order: BUY TGT at \$86.90, Orig Vol: 230, Rem Vol: 230, Fill Vol: 0, CXL Vol: 0, ID: EGGTGT\$86.9082076807646800
- 9) ADD: BUY: CAT order: BUY TGT at \$87.75, Orig Vol: 205, Rem Vol: 205, Fill Vol: 0, CXL Vol: 0, ID: CATTGT\$87.7582076807924000
- PARTIAL FILL: (SELL 205) CAT order: SELL TGT at \$87.60, Orig Vol: 300, Rem Vol: 95, Fill Vol: 205, CXL Vol: 0, ID: CATTGT\$87.6082076753773700*
- FILL: (BUY 205) CAT order: BUY TGT at \$87.75, Orig Vol: 205, Rem Vol: 0, Fill Vol: 205, CXL Vol: 0, ID: CATTGT\$87.7582076807924000*
- 10) ADD: BUY: CAT order: BUY TGT at \$87.05, Orig Vol: 105, Rem Vol: 105, Fill Vol: 0, CXL Vol: 0, ID: CATTGT\$87.0582076815280600

11) CANCEL: SELL Order: BOBTSLA\$99.1582076806203800 Cxl Qty: 210  
12) ADD: SELL: EGG order: SELL TGT at \$88.00, Orig Vol: 25, Rem Vol: 25, Fill Vol: 0, CXL Vol: 0,  
ID: EGGTGT\$88.0082076819821100  
13) ADD: BUY: CAT order: BUY AMZN at \$50.80, Orig Vol: 115, Rem Vol: 115, Fill Vol: 0, CXL Vol: 0,  
ID: CATAMZN\$50.8082076820169300  
14) ADD: BUY: EGG order: BUY TGT at \$87.15, Orig Vol: 295, Rem Vol: 295, Fill Vol: 0, CXL Vol: 0,  
ID: EGGTGT\$87.1582076820461500  
15) ADD: SELL: BOB order: SELL TSLA at \$97.90, Orig Vol: 210, Rem Vol: 210, Fill Vol: 0, CXL Vol: 0,  
ID: BOBTSLA\$97.9082076820717200  
PARTIAL FILL: (SELL 140) BOB order: SELL TSLA at \$97.90, Orig Vol: 210, Rem Vol: 70, Fill Vol: 140,  
CXL Vol: 0, ID: BOBTSLA\$97.9082076820717200  
FILL: (BUY 140) DOG order: BUY TSLA at \$98.45, Orig Vol: 140, Rem Vol: 0, Fill Vol: 140, CXL Vol:  
0, ID: DOGTSLA\$98.4582076807364900  
16) ADD: BUY: CAT order: BUY WMT at \$70.65, Orig Vol: 175, Rem Vol: 175, Fill Vol: 0, CXL Vol: 0,  
ID: CATWMT\$70.6582076821249200  
17) ADD: BUY: BOB order: BUY AMZN at \$51.20, Orig Vol: 155, Rem Vol: 155, Fill Vol: 0, CXL Vol: 0,  
ID: BOBAMZN\$51.2082076821468600  
FILL: (SELL 45) ANN order: SELL AMZN at \$50.85, Orig Vol: 45, Rem Vol: 0, Fill Vol: 45, CXL Vol: 0,  
ID: ANNAMZN\$50.8582076805829000  
PARTIAL FILL: (BUY 45) BOB order: BUY AMZN at \$51.20, Orig Vol: 155, Rem Vol: 110, Fill Vol: 45,  
CXL Vol: 0, ID: BOBAMZN\$51.2082076821468600  
18) ADD: BUY: EGG order: BUY TSLA at \$98.10, Orig Vol: 180, Rem Vol: 180, Fill Vol: 0, CXL Vol: 0,  
ID: EGGTSLA\$98.1082076821946900  
FILL: (SELL 70) BOB order: SELL TSLA at \$97.90, Orig Vol: 210, Rem Vol: 0, Fill Vol: 210, CXL Vol:  
0, ID: BOBTSLA\$97.9082076820717200  
PARTIAL FILL: (BUY 70) EGG order: BUY TSLA at \$98.10, Orig Vol: 180, Rem Vol: 110, Fill Vol: 70,  
CXL Vol: 0, ID: EGGTSLA\$98.1082076821946900  
19) ADD: BUY: ANN order: BUY WMT at \$69.90, Orig Vol: 160, Rem Vol: 160, Fill Vol: 0, CXL Vol: 0,  
ID: ANNWMT\$69.9082076822482500  
20) ADD: SELL: CAT order: SELL WMT at \$71.05, Orig Vol: 115, Rem Vol: 115, Fill Vol: 0, CXL Vol: 0,  
ID: CATWMT\$71.0582076822738000  
FILL: (SELL 115) CAT order: SELL WMT at \$71.05, Orig Vol: 115, Rem Vol: 0, Fill Vol: 115, CXL Vol:  
0, ID: CATWMT\$71.0582076822738000  
PARTIAL FILL: (BUY 115) BOB order: BUY WMT at \$71.15, Orig Vol: 175, Rem Vol: 60, Fill Vol: 115,  
CXL Vol: 0, ID: BOBWMT\$71.1582076807077000  
21) CANCEL: BUY Order: EGGTSLA\$98.1082076821946900 Cxl Qty: 110  
22) ADD: SELL: DOG order: SELL AMZN at \$51.10, Orig Vol: 60, Rem Vol: 60, Fill Vol: 0, CXL Vol: 0,  
ID: DOGAMZN\$51.1082076823357800  
FILL: (SELL 60) DOG order: SELL AMZN at \$51.10, Orig Vol: 60, Rem Vol: 0, Fill Vol: 60, CXL Vol: 0,  
ID: DOGAMZN\$51.1082076823357800  
PARTIAL FILL: (BUY 60) BOB order: BUY AMZN at \$51.20, Orig Vol: 155, Rem Vol: 50, Fill Vol: 105,  
CXL Vol: 0, ID: BOBAMZN\$51.2082076821468600  
23) CANCEL: BUY Order: CATAMZN\$50.8082076820169300 Cxl Qty: 115  
24) ADD: SELL: DOG order: SELL WMT at \$70.75, Orig Vol: 65, Rem Vol: 65, Fill Vol: 0, CXL Vol: 0,  
ID: DOGWMT\$70.7582076823983000  
PARTIAL FILL: (SELL 60) DOG order: SELL WMT at \$70.75, Orig Vol: 65, Rem Vol: 5, Fill Vol: 60, CXL  
Vol: 0, ID: DOGWMT\$70.7582076823983000  
FILL: (BUY 60) BOB order: BUY WMT at \$71.15, Orig Vol: 175, Rem Vol: 0, Fill Vol: 175, CXL Vol: 0,  
ID: BOBWMT\$71.1582076807077000  
25) ADD: BUY: CAT order: BUY TSLA at \$98.75, Orig Vol: 290, Rem Vol: 290, Fill Vol: 0, CXL Vol: 0,  
ID: CATTSLA\$98.7582076824539800  
26) ADD: BUY: CAT order: BUY WMT at \$70.45, Orig Vol: 240, Rem Vol: 240, Fill Vol: 0, CXL Vol: 0,  
ID: CATWMT\$70.4582076824757300  
27) ADD: BUY: ANN order: BUY TSLA at \$97.80, Orig Vol: 255, Rem Vol: 255, Fill Vol: 0, CXL Vol: 0,  
ID: ANNTSLA\$97.8082076824966800  
28) CANCEL: BUY Order: EGGTGT\$87.1582076820461500 Cxl Qty: 295  
29) ADD: SELL: CAT order: SELL WMT at \$69.80, Orig Vol: 260, Rem Vol: 260, Fill Vol: 0, CXL Vol: 0,  
ID: CATWMT\$69.8082076825243200  
FILL: (SELL 260) CAT order: SELL WMT at \$69.80, Orig Vol: 260, Rem Vol: 0, Fill Vol: 260, CXL Vol:  
0, ID: CATWMT\$69.8082076825243200  
FILL: (BUY 195) BOB order: BUY WMT at \$70.65, Orig Vol: 195, Rem Vol: 0, Fill Vol: 195, CXL Vol: 0,  
ID: BOBWMT\$70.6582076806796500



PARTIAL FILL: (BUY 65) CAT order: BUY WMT at \$70.65, Orig Vol: 175, Rem Vol: 110, Fill Vol: 65, CXL Vol: 0, ID: CATWMT\$70.6582076821249200

30) ADD: BUY: BOB order: BUY AMZN at \$51.20, Orig Vol: 105, Rem Vol: 105, Fill Vol: 0, CXL Vol: 0, ID: BOBAMZN\$51.2082076826078800

31) ADD: SELL: ANN order: SELL WMT at \$70.90, Orig Vol: 65, Rem Vol: 65, Fill Vol: 0, CXL Vol: 0, ID: ANNWMT\$70.9082076826311400

32) CANCEL: BUY Order: ANNWMT\$69.9082076822482500 Cxl Qty: 160

33) ADD: BUY: CAT order: BUY TGT at \$87.30, Orig Vol: 235, Rem Vol: 235, Fill Vol: 0, CXL Vol: 0, ID: CATTGT\$87.3082076826613700

34) ADD: SELL: EGG order: SELL WMT at \$70.84, Orig Vol: 30, Rem Vol: 30, Fill Vol: 0, CXL Vol: 0, ID: EGGWMT\$70.8482076826918700

35) ADD: BUY: BOB order: BUY AMZN at \$50.95, Orig Vol: 140, Rem Vol: 140, Fill Vol: 0, CXL Vol: 0, ID: BOBAMZN\$50.9582076827139700

36) ADD: SELL: ANN order: SELL TSLA at \$98.90, Orig Vol: 120, Rem Vol: 120, Fill Vol: 0, CXL Vol: 0, ID: ANNTSLA\$98.9082076827372700

37) ADD: SELL: EGG order: SELL TGT at \$87.30, Orig Vol: 150, Rem Vol: 150, Fill Vol: 0, CXL Vol: 0, ID: EGGTGT\$87.3082076827620100

FILL: (SELL 150) EGG order: SELL TGT at \$87.30, Orig Vol: 150, Rem Vol: 0, Fill Vol: 150, CXL Vol: 0, ID: EGGTGT\$87.3082076827620100

PARTIAL FILL: (BUY 150) CAT order: BUY TGT at \$87.30, Orig Vol: 235, Rem Vol: 85, Fill Vol: 150, CXL Vol: 0, ID: CATTGT\$87.3082076826613700

38) ADD: BUY: CAT order: BUY WMT at \$69.90, Orig Vol: 105, Rem Vol: 105, Fill Vol: 0, CXL Vol: 0, ID: CATWMT\$69.9082076828178100

39) ADD: SELL: BOB order: SELL TSLA at \$98.30, Orig Vol: 250, Rem Vol: 250, Fill Vol: 0, CXL Vol: 0, ID: BOBTSLA\$98.3082076828447800

FILL: (SELL 250) BOB order: SELL TSLA at \$98.30, Orig Vol: 250, Rem Vol: 0, Fill Vol: 250, CXL Vol: 0, ID: BOBTSLA\$98.3082076828447800

PARTIAL FILL: (BUY 250) CAT order: BUY TSLA at \$98.75, Orig Vol: 290, Rem Vol: 40, Fill Vol: 250, CXL Vol: 0, ID: CATTSLA\$98.7582076824539800

40) ADD: SELL: EGG order: SELL WMT at \$70.25, Orig Vol: 190, Rem Vol: 190, Fill Vol: 0, CXL Vol: 0, ID: EGGWMT\$70.2582076828958300

PARTIAL FILL: (SELL 110) EGG order: SELL WMT at \$70.25, Orig Vol: 190, Rem Vol: 80, Fill Vol: 110, CXL Vol: 0, ID: EGGWMT\$70.2582076828958300

FILL: (BUY 110) CAT order: BUY WMT at \$70.65, Orig Vol: 175, Rem Vol: 0, Fill Vol: 175, CXL Vol: 0, ID: CATWMT\$70.6582076821249200

FILL: (SELL 80) EGG order: SELL WMT at \$70.25, Orig Vol: 190, Rem Vol: 0, Fill Vol: 190, CXL Vol: 0, ID: EGGWMT\$70.2582076828958300

PARTIAL FILL: (BUY 80) CAT order: BUY WMT at \$70.45, Orig Vol: 240, Rem Vol: 160, Fill Vol: 80, CXL Vol: 0, ID: CATWMT\$70.4582076824757300

---

ProductBooks:

Product: TGT

Side: BUY

Price: \$87.30

CAT order: BUY TGT at \$87.30, Orig Vol: 235, Rem Vol: 85, Fill Vol: 150, CXL Vol: 0, ID: CATTGT\$87.3082076826613700

Price: \$87.05

CAT order: BUY TGT at \$87.05, Orig Vol: 105, Rem Vol: 105, Fill Vol: 0, CXL Vol: 0, ID: CATTGT\$87.0582076815280600

Price: \$86.90

EGG order: BUY TGT at \$86.90, Orig Vol: 230, Rem Vol: 230, Fill Vol: 0, CXL Vol: 0, ID: EGGTGT\$86.9082076807646800

Side: SELL

Price: \$87.60

CAT order: SELL TGT at \$87.60, Orig Vol: 300, Rem Vol: 95, Fill Vol: 205, CXL Vol: 0, ID: CATTGT\$87.6082076753773700

Price: \$88.00

EGG order: SELL TGT at \$88.00, Orig Vol: 25, Rem Vol: 25, Fill Vol: 0, CXL Vol: 0, ID: EGGTGT\$88.0082076819821100

```
Product: TSLA
Side: BUY
  Price: $98.75
    CAT order: BUY TSLA at $98.75, Orig Vol: 290, Rem Vol: 40, Fill Vol: 250, CXL Vol: 0,
               ID: CATTSLA$98.7582076824539800
  Price: $97.80
    ANN order: BUY TSLA at $97.80, Orig Vol: 255, Rem Vol: 255, Fill Vol: 0, CXL Vol: 0, ID:
               ANNTSLA$97.8082076824966800
  Price: $97.75
    EGG order: BUY TSLA at $97.75, Orig Vol: 240, Rem Vol: 240, Fill Vol: 0, CXL Vol: 0, ID:
               EGGTSLA$97.7582076806500700
Side: SELL
  Price: $98.90
    ANN order: SELL TSLA at $98.90, Orig Vol: 120, Rem Vol: 120, Fill Vol: 0, CXL Vol: 0,
               ID: ANNTSLA$98.9082076827372700

Product: AMZN
Side: BUY
  Price: $51.20
    BOB order: BUY AMZN at $51.20, Orig Vol: 155, Rem Vol: 50, Fill Vol: 105, CXL Vol: 0,
               ID: BOBAMZN$51.2082076821468600
    BOB order: BUY AMZN at $51.20, Orig Vol: 105, Rem Vol: 105, Fill Vol: 0, CXL Vol: 0, ID:
               BOBAMZN$51.2082076826078800
  Price: $50.95
    BOB order: BUY AMZN at $50.95, Orig Vol: 140, Rem Vol: 140, Fill Vol: 0, CXL Vol: 0, ID:
               BOBAMZN$50.9582076827139700
Side: SELL
  <Empty>

Product: WMT
Side: BUY
  Price: $70.45
    CAT order: BUY WMT at $70.45, Orig Vol: 240, Rem Vol: 160, Fill Vol: 80, CXL Vol: 0, ID:
               CATWMT$70.4582076824757300
  Price: $69.90
    CAT order: BUY WMT at $69.90, Orig Vol: 105, Rem Vol: 105, Fill Vol: 0, CXL Vol: 0, ID:
               CATWMT$69.9082076828178100
Side: SELL
  Price: $70.75
    DOG order: SELL WMT at $70.75, Orig Vol: 65, Rem Vol: 5, Fill Vol: 60, CXL Vol: 0, ID:
               DOGWMT$70.7582076823983000
  Price: $70.84
    EGG order: SELL WMT at $70.84, Orig Vol: 30, Rem Vol: 30, Fill Vol: 0, CXL Vol: 0, ID:
               EGGWMT$70.8482076826918700
  Price: $70.90
    ANN order: SELL WMT at $70.90, Orig Vol: 65, Rem Vol: 65, Fill Vol: 0, CXL Vol: 0, ID:
               ANNWMT$70.9082076826311400
```

---

Users:

User Id: ANN

```
Product: WMT, Price: $69.90, OriginalVolume: 160, RemainingVolume: 0, CancelledVolume: 160,
  FilledVolume: 0, User: ANN, Side: BUY, Id: ANNWMT$69.9082076822482500
Product: WMT, Price: $70.90, OriginalVolume: 65, RemainingVolume: 65, CancelledVolume: 0,
  FilledVolume: 0, User: ANN, Side: SELL, Id: ANNWMT$70.9082076826311400
Product: TSLA, Price: $97.80, OriginalVolume: 255, RemainingVolume: 255, CancelledVolume: 0,
  FilledVolume: 0, User: ANN, Side: BUY, Id: ANNTSLA$97.8082076824966800
Product: AMZN, Price: $50.85, OriginalVolume: 45, RemainingVolume: 0, CancelledVolume: 0,
  FilledVolume: 45, User: ANN, Side: SELL, Id: ANNAMZN$50.8582076805829000
```



Product: TSLA, Price: \$98.90, OriginalVolume: 120, RemainingVolume: 120, CancelledVolume: 0,  
FilledVolume: 0, User: ANN, Side: SELL, Id: ANNTSLA\$98.9082076827372700

User Id: BOB

Product: AMZN, Price: \$51.20, OriginalVolume: 155, RemainingVolume: 50, CancelledVolume: 0,  
FilledVolume: 105, User: BOB, Side: BUY, Id: BOBAMZN\$51.2082076821468600  
Product: TSLA, Price: \$99.15, OriginalVolume: 210, RemainingVolume: 0, CancelledVolume: 210,  
FilledVolume: 0, User: BOB, Side: SELL, Id: BOBTSLA\$99.1582076806203800  
Product: AMZN, Price: \$51.20, OriginalVolume: 105, RemainingVolume: 105, CancelledVolume: 0,  
FilledVolume: 0, User: BOB, Side: BUY, Id: BOBAMZN\$51.2082076826078800  
Product: TSLA, Price: \$98.30, OriginalVolume: 250, RemainingVolume: 250, CancelledVolume: 0,  
FilledVolume: 0, User: BOB, Side: SELL, Id: BOBTSLA\$98.3082076828447800  
Product: WMT, Price: \$71.15, OriginalVolume: 175, RemainingVolume: 0, CancelledVolume: 0,  
FilledVolume: 175, User: BOB, Side: BUY, Id: BOBWMT\$71.1582076807077000  
Product: TSLA, Price: \$97.90, OriginalVolume: 210, RemainingVolume: 0, CancelledVolume: 0,  
FilledVolume: 210, User: BOB, Side: SELL, Id: BOBTSLA\$97.9082076820717200  
Product: AMZN, Price: \$50.95, OriginalVolume: 140, RemainingVolume: 140, CancelledVolume: 0,  
FilledVolume: 0, User: BOB, Side: BUY, Id: BOBAMZN\$50.9582076827139700  
Product: WMT, Price: \$70.65, OriginalVolume: 195, RemainingVolume: 0, CancelledVolume: 0,  
FilledVolume: 195, User: BOB, Side: BUY, Id: BOBWMT\$70.6582076806796500

User Id: EGG

Product: TSLA, Price: \$98.10, OriginalVolume: 180, RemainingVolume: 0, CancelledVolume: 110,  
FilledVolume: 70, User: EGG, Side: BUY, Id: EGGTSLA\$98.1082076821946900  
Product: TGT, Price: \$87.15, OriginalVolume: 295, RemainingVolume: 0, CancelledVolume: 295,  
FilledVolume: 0, User: EGG, Side: BUY, Id: EGGTGT\$87.1582076820461500  
Product: WMT, Price: \$70.84, OriginalVolume: 30, RemainingVolume: 30, CancelledVolume: 0,  
FilledVolume: 0, User: EGG, Side: SELL, Id: EGGWMT\$70.8482076826918700  
Product: TGT, Price: \$87.30, OriginalVolume: 150, RemainingVolume: 150, CancelledVolume: 0,  
FilledVolume: 0, User: EGG, Side: SELL, Id: EGGTGT\$87.3082076827620100  
Product: WMT, Price: \$70.25, OriginalVolume: 190, RemainingVolume: 190, CancelledVolume: 0,  
FilledVolume: 0, User: EGG, Side: SELL, Id: EGGWMT\$70.2582076828958300  
Product: TGT, Price: \$88.00, OriginalVolume: 25, RemainingVolume: 25, CancelledVolume: 0,  
FilledVolume: 0, User: EGG, Side: SELL, Id: EGGTGT\$88.0082076819821100  
Product: TGT, Price: \$86.90, OriginalVolume: 230, RemainingVolume: 230, CancelledVolume: 0,  
FilledVolume: 0, User: EGG, Side: BUY, Id: EGGTGT\$86.9082076807646800  
Product: TSLA, Price: \$97.75, OriginalVolume: 240, RemainingVolume: 240, CancelledVolume: 0,  
FilledVolume: 0, User: EGG, Side: BUY, Id: EGGTSLA\$97.7582076806500700

User Id: CAT

Product: AMZN, Price: \$50.80, OriginalVolume: 115, RemainingVolume: 0, CancelledVolume: 115,  
FilledVolume: 0, User: CAT, Side: BUY, Id: CATAMZN\$50.8082076820169300  
Product: WMT, Price: \$70.45, OriginalVolume: 240, RemainingVolume: 160, CancelledVolume: 0,  
FilledVolume: 80, User: CAT, Side: BUY, Id: CATWMT\$70.4582076824757300  
Product: TGT, Price: \$87.60, OriginalVolume: 300, RemainingVolume: 95, CancelledVolume: 0,  
FilledVolume: 205, User: CAT, Side: SELL, Id: CATTGT\$87.6082076753773700  
Product: TGT, Price: \$87.30, OriginalVolume: 235, RemainingVolume: 85, CancelledVolume: 0,  
FilledVolume: 150, User: CAT, Side: BUY, Id: CATTGT\$87.3082076826613700  
Product: TSLA, Price: \$98.75, OriginalVolume: 290, RemainingVolume: 40, CancelledVolume: 0,  
FilledVolume: 250, User: CAT, Side: BUY, Id: CATTSLA\$98.7582076824539800  
Product: WMT, Price: \$69.80, OriginalVolume: 260, RemainingVolume: 260, CancelledVolume: 0,  
FilledVolume: 0, User: CAT, Side: SELL, Id: CATWMT\$69.8082076825243200  
Product: WMT, Price: \$70.65, OriginalVolume: 175, RemainingVolume: 0, CancelledVolume: 0,  
FilledVolume: 175, User: CAT, Side: BUY, Id: CATWMT\$70.6582076821249200  
Product: WMT, Price: \$69.90, OriginalVolume: 105, RemainingVolume: 105, CancelledVolume: 0,  
FilledVolume: 0, User: CAT, Side: BUY, Id: CATWMT\$69.9082076828178100  
Product: TGT, Price: \$87.05, OriginalVolume: 105, RemainingVolume: 105, CancelledVolume: 0,  
FilledVolume: 0, User: CAT, Side: BUY, Id: CATTGT\$87.0582076815280600  
Product: WMT, Price: \$71.05, OriginalVolume: 115, RemainingVolume: 115, CancelledVolume: 0,  
FilledVolume: 0, User: CAT, Side: SELL, Id: CATWMT\$71.0582076822738000  
Product: TGT, Price: \$87.75, OriginalVolume: 205, RemainingVolume: 205, CancelledVolume: 0,  
FilledVolume: 0, User: CAT, Side: BUY, Id: CATTGT\$87.7582076807924000

User Id: DOG

Product: WMT, Price: \$70.75, OriginalVolume: 65, RemainingVolume: 65, CancelledVolume: 0,  
FilledVolume: 0, User: DOG, Side: SELL, Id: DOGWMT\$70.7582076823983000

Product: TSLA, Price: \$98.45, OriginalVolume: 140, RemainingVolume: 0, CancelledVolume: 0,  
FilledVolume: 140, User: DOG, Side: BUY, Id: DOGTSLA\$98.4582076807364900

Product: AMZN, Price: \$51.10, OriginalVolume: 60, RemainingVolume: 60, CancelledVolume: 0,  
FilledVolume: 0, User: DOG, Side: SELL, Id: DOGAMZN\$51.1082076823357800

### Project Assistance

If you are stuck on some design or code related problem *that you have exhaustively researched and/or debugged yourself*, you can email me a ZIP file of your entire project so that I can examine the problem. All emailed assistance requests must include a detailed description of the problem, and the details of what steps you have already taken in trying to determine the source of the problem.

### Submissions & Grading

When submitting, you should submit a ZIP file of your entire project so that I can compile and execute it on my end. To reduce the size of the zipped file, please remove the “out” folder from your project (if present) before zipping. (This folder is automatically regenerated each time you compile/run so it's safe to delete)

The following are the key points that will be examined in Project when graded:

- Good object-oriented design & implementation
- Proper use of java language conventions
- Proper object-oriented coding practices
- Correct, accurate application execution

Submissions must reflect the concepts and practices we cover in class, and the requirements specified in this document.

Please review the Academic Integrity and Plagiarism section of the syllabus before, during, and after working on this assignment. All work must be your own.

Late submission up to 1 week late will be accepted, though late submissions will incur a 10% penalty (i.e., 10 points). *No late submissions will be accepted for grading after that 1-week time period has passed.*

*If you do not understand anything in this handout, please ask. Otherwise the assumption is that you understand the content.*

**Appendix A: “runSim” method**


**Appendix B: “getPrice” method**
