**BrokerageAccount Subclass Exercise**

**Directions:** Create the concrete classes below. The Stock and BrokerageAccount classes have an A*ggregation* relationship (the BrokerageAccount class is created with two Stock instance variables). The BrokerageAccount and Account classes share an inheritance relationship with BrokerageAccount extending Account.

**Account**

**Instance Variables**

* Strings for the account number and name of the account owner
* Double for the balance of the account

**Constructors**

* *new Account()* – initializes each instance variable to standard default value
* *new Account(“James Bond”, 007007)*
* *new Account(“James Bond”, 007007, 0.0)*

**Methods**

* Create mutators and accessors for each instance variable
* *toString()* – of the form:

Account:

Owner: Crone

Account Number: 123123123

Balance: $0.0

**Stock**

**Instance Variables**

* String for the symbol of the stock
* Double for the price of the stock
* Integer for the quantity of the stock

**Constructors**

* *new Stock()* – initializes each instance variable to standard default value
* *new Stock(“KO”, 40.44, 3)*

**Methods**

* Create mutators and accessors for each instance variable
* *getStockValue()*  - returns the price of the stock multiplied by the quantity
* *toString()* – of the form:

3 shares of KO at $40.0

**BrokerageAccount**

**Instance Variables**

* A List that will hold currently owned stocks

\*\* Must have type parameter

**Constructors**

\*\*Each constructor below must also instantiate the List of stocks AND call *super*

* *new BrokerageAccount() -* initializes each instance variable to standard default value
* *new BrokerageAccount(“Bond”, 007007)*
* *new BrokerageAccount(“Bond”, 007007, 500.0)*

**Methods**

* *getStockList()*  - returns the list of currently owned stocks
* *getTotalStockValue()* – returns a double value that represents the value of all of the currently owned stocks
* *buyStock(Stock s)* – returns false if there is not enough money inthe account balance. Returns true and deducts from balance if the transaction is successful. Adds the purchased stock to the list of stocks currently owned.
* *sellStock(String symbol)* – sells ALL of the stocks held with the input symbol. Returns false if the symbol is not found. Returns true and updates balance if the stocks are sold.
* *toString()* – of the form:

Brokerage Account:

Owner: Crone

Account Number: 123123123

Balance: $500.0

Stocks: []

**AccountMain**

**Directions:**

* 1. Instantiate a new Account *a1* with your name and any account number
  2. Print *a1*
  3. Instantiate a new BrokerageAccount *b1* with your name, any account number, and a starting value of $500.00.
  4. Print *b1*
  5. Print the result of buying 3 shares of KO at $40/share.
  6. Print the new account balance of b1.
  7. Print the result of buying 3 shares of CVX at $85/share.
  8. Print the new account balance of b1.
  9. Print the result of buying 5 shares of PPL at $45/share.
  10. Print the new account balance of b1.
  11. Print the currently owned stocks.
  12. Print the total value of all of the funds in the brokerage account.
  13. Print the result of selling KO.
  14. Print the new account balance of b1.
  15. Print the result of selling CVX.
  16. Print the new account balance of b1.
  17. Print the result of selling PPL.
  18. Print the new account balance of b1.
  19. Print the currently owned stocks.

A Correct Printout)

Account:

Owner: Crone

Account Number: 123123123

Balance: $0.0

Brokerage Account:

Owner: Crone

Account Number: 123123123

Balance: $500.0

Stocks: []

Buy 3 Shares of KO at $40: true

New balance: $380.0

Buy 3 Shares of CVX at $85: true

New balance: $125.0

Buy 5 Shares of PPL at $45: false

New balance: $125.0

Stocks currently owned: [3 shares of KO at $40.0, 3 shares of CVX at $85.0]

Total Brokerage Account Value: $500.0

Sell KO: true

New balance: $245.0

Sell CVX: true

New balance: $500.0

Sell PPL: false

New balance: $500.0

Stocks currently owned: []