HW1

August 14, 2024

[18]:
```
import random

class Portfolio:
    def __init__(self, name):
        self.name = name
        self.cash = 0
        self.stocks = {}
        self.mutual_funds = {}
        self.transaction_history = []

    def addCash(self, amount):
        self.cash += amount
        self.transaction_history.append(f"Added ${amount:.2f} cash.")

    def withdrawCash(self, amount):
        if self.cash >= amount:
            self.cash -= amount
            self.transaction_history.append(f"Withdrew ${amount:.2f} cash.")
        else:
            print("Insufficient cash.")

    def buyStock(self, quantity, stock):
        total_cost = quantity * stock.price
        if self.cash >= total_cost:
            self.cash -= total_cost
            if stock.symbol in self.stocks:
                self.stocks[stock.symbol]['quantity'] += quantity
            else:
                self.stocks[stock.symbol] = {'quantity': quantity, 'price':␣
 ↪stock.price}
            self.transaction_history.append(f"Bought {quantity} shares of␣
 ↪{stock.symbol} stock at ${stock.price:.2f} per share.")
        else:
            print("Insufficient cash.")

    def sellStock(self, symbol, quantity):
```

```
        if symbol in self.stocks and self.stocks[symbol]['quantity'] >=␣
↪quantity:
            original_price = self.stocks[symbol]['price']
            sell_price = random.uniform(0.5 * original_price, 1.5 *␣
↪original_price)
            self.cash += quantity * sell_price
            self.stocks[symbol]['quantity'] -= quantity
            if self.stocks[symbol]['quantity'] == 0:
                del self.stocks[symbol]
            self.transaction_history.append(f"Sold {quantity} shares of␣
↪{symbol} stock at ${sell_price:.2f} per share.")
        else:
            print("Insufficient shares to sell.")

    def buyMutualFund(self, quantity, mf):
        total_cost = quantity * 1  # Mutual Funds are always $1 per share
        if self.cash >= total_cost:
            self.cash -= total_cost
            if mf.symbol in self.mutual_funds:
                self.mutual_funds[mf.symbol] += quantity
            else:
                self.mutual_funds[mf.symbol] = quantity
            self.transaction_history.append(f"Bought {quantity} shares of {mf.
↪symbol} mutual fund.")
        else:
            print("Insufficient cash to buy mutual fund.")

    def sellMutualFund(self, symbol, quantity):
        if symbol in self.mutual_funds and self.mutual_funds[symbol] >=␣
↪quantity:
            sell_price = random.uniform(0.9, 1.2)
            self.cash += quantity * sell_price
            self.mutual_funds[symbol] -= quantity
            if self.mutual_funds[symbol] == 0:
                del self.mutual_funds[symbol]
            self.transaction_history.append(f"Sold {quantity} shares of␣
↪{symbol} mutual fund at ${sell_price:.2f} per share.")
        else:
            print("Insufficient shares to sell.")

    def history(self):
        for transaction in self.transaction_history:
            print(transaction)

    def __str__(self):
```

```
        stocks_str = ", ".join([f"{v['quantity']} {k}" for k, v in self.stocks.
↪items()])
        mfs_str = ", ".join([f"{v} {k}" for k, v in self.mutual_funds.items()])
        return f"Portfolio '{self.name}'\nCash: ${self.cash:.2f}\nStocks:␣
↪{stocks_str}\nMutual Funds: {mfs_str}"

class Stock:
    def __init__(self, price, symbol):
        self.price = price
        self.symbol = symbol

class MutualFund:
    def __init__(self, symbol):
        self.symbol = symbol
```

[19]:

```
Added $300.50 cash.
Bought 5 shares of HFH stock at $20.00 per share.
Sold 1 shares of HFH stock at $18.44 per share.
Withdrew $50.00 cash.
```