MQF Spring 2019

Homework 3

Due 3/3/2019 before midnight

Write a Python program which will create at least three classes to perform financial management.

A portfolio contains assets of a person, an organization, or a company. An asset can be cash in an account, real estate, or stocks. To simplify our program, assume portfolio has a name as its attribute.

An account contains an attribute balance. Money can be deposited into this account, and can be withdrew from this account and can be transferred from one account to another account.

A real estate contains an address_ID, and current market value. Although the market value can be modified, in our simulation here, we don't change its value. A portfolio may contain zero or more real estate. New real estate can be added and existing one can be sold using market value. The sold money will be added to its account. To buy, make sure there is enough fund to buy. Balance of fund need to be updated. If not enough fund to buy, an exception must be raised.

A portfolio may contain zero or more stocks. For each stock, there is stock symbol, number of shares and total cost. To simplify the computation if one sells x shares of this stock, the total cost can be updated as the total cost subtract (total_cost /number_of_shares) * x. If one buys x shares with y dollars each share then number_of_shares += x, and total_cost += (x * y). Note you must have enough fund to buy stock. If not enough fund, an exception must be raised.

In addition to the individual portfolio manipulation, it is possible for two portfolios to be merged together. Transfer of fund from one portfolio to another can be done as well. Real estate and stocks can be transferred too. The price of stock will use simple method mentioned above. During stock buying and selling, balances of accounts (to and from portfolio's) may need to be changed.

Write a main program which will perform the creation of portfolios, transaction, and then output the simulation results of these portfolios' result to a text file. The output format of all portfolios shall include the name of portfolio, account balance, all stocks (each stock symbol, shares, total_cost), each real estate with address_ID, and its value.

To simplify the simulation by using an input file which contains all creations of portfolio and transactions.

The format would look like the following:

Cp : create portfolio	Cp, name_of_portfolio
Bs: buy stock	Bs, stock_symbol, number_of_share, purchase_cost_each_share, name_of_portfolio
Ss: sell stock	Ss, stock_symbol, number_of_share, sell_price_each_share, name_of_portfolio
Mg: merge two portfolios	Mg, portfolio_name_1, portfolio_name_2, result_name
Br: buy real estate	Br, address_id, buying_price, name_of_portfolio
Sr: sell real estate	Sr, address_id, sell_price, name_of_portfolio
Xs: transfer stock	Xs, source_portfolio_name, destination_portfolio_name, stock_name, shares
Xr: transfer real estate	Xr, source_portfolio_name, destination_portfolio_name, real_estate_address_id
Xf: transfer fund	Xf, source_portfolio_name, destination_portfolio_name, amount_to_transfer
Dp: deposit money to account	Dp, portfolio_name, amount_to_deposit
Wd: withdraw money from account	Wd, portfolio_name, amount_to_withdraw

For example:

Cp, prudential_1 # create a portfolio name prudential_1

Dp, prudential_1, 1000000 #deposit 1 million dollars to prudential_1

Bs, MSFT, 100, 58.32, prudential_1 # buy Microsoft stock 100 shares with each share $58.32. for prudential _1. make sure there is enough fund on prudential_1's account

Cp, prudential_2 # create a portfolio name prudential_2

Dp, prudential_2, 1000000000 #deposit 1 billion dollars to prudential_2

Br, Newark_NJ_1400, 500000 # buy a piece of building from Newark_NJ_1400 with price 500 thousands. Make sure that you have enough money in your account to buy

Br, Morris_NJ_1379, 1500000 # buy a piece of building from Morris_NJ_1379 with price one million 500 thousand dollars. Make sure that you have enough money in your account to buy

Wd, prudential_2, 1000000 # withdraw one million dollars from prudential_2

Xf, prudential_2, prudential_1, 100.00 # transfer one hundred dollars from prudential_1 to prudential_2

Xs, prudential_1, prudential_2, MSFT, 20 #transfer 20 shares of MSFT stock from prudential_1 to prudential_2

Xr, prudential_1, prudential_2, Morris_NJ_1379 # transfer property Morris_NJ_1379 from prudential_1 to prudential_2

Mg, prudential_1, prudential_2, PRUD # merge prudential_1, prudential_2 to become PRUD

Your TA will give you a test case later. You shall make sure to test more cases. Your TA may use different test case in grading.

Requirements:

Must include at least three Python classes. Overload at least operator __str__ so that we can print the object of that type of class.

Submission

Total 60 points

Correctly implement of class (at least three classes)	30 points
Use files correctly (output files)	5 points
Exception handling	5 points
Correctness of your program output	20 points
Submit to Blackboard	

Due 3/3/2019 before midnight. 10% penalty for each late day thereafter.