Lab11

Suppose we need to extend the functionality of `BankAccount` class of previous lab such that we are able to maintain history of transactions; i.e. withdraw and deposit performed on an account. To implement this, let us say we have identified a new class called `Transaction` – a transaction object holds three values- `Date`, `Transaction-Type`, `Amount`., and balance in account after the transaction. `Transaction-Type` is a short value (1 and 2), where 1 is for deposit and 2 for withdrawal. To store transaction history, let us say, an bank account account object holds ordered `collection of Transaction` objects.

For `Date` you can use date class created in one of the past lab.

You are required to do following-

1. Provide implementation of `Transaction` class. Implement methods: (a) constructor that initializes required fields and accessors for required fields (let us say no mutator method is required).
2. Modify and implement Bank Account class for added functionality. Do not change anything that make any of client codes of Lab10 failing.
3. Also add following methods
    1. `averageMonthlyBalance(string month)` to the `BankAccount` class that returns average balance given month. Let month parameter be of type string in the format of `"yy-mm"`.
       This method computes and returns weighted average balance over all transactions on the account. For example, in an account balance of 5500 remains for 9 days, 3400 for 21 days, then the average balance = (5500*9+3400*21)/30.
    2. `vector<Transaction> getTransactions(Date from, Date to)` that returns a vector of transaction objects for a given date range.
    3. Overload `operator=` for Bank Account class. Ensure Deep Copying.
4. Do testing of Bank Account class as following
    1. Run both client codes of `Lab10` using newer bank account class, a make sure that none of the code crashes.
    2. Create a new client code that creates a bank account object; adds few transactions, and print
        1. Passbook for a given duration (date range)
        2. Average balance of the account for a given month.
    3. Create a new client code that creates a **Saving bank account** object; adds few transactions, and print
        1. Passbook for a given duration (date range)
        2. Average balance of the account for a given month.
        3. [optional] Interest paid during given month