STOCK MANAGEMENT SYSTEM

NAME: DIKSHA PRAKASH
NET-ID: dp978

This project is a pseudo-implementation of the stock market account management system which allows the users to interact with the stock portfolio, buy and sell shares and also maintain a bank account associated with it. One can access multiple details regarding the bank account and stock portfolio at one go.

- All the requirements of problem statement are satisfied in the project (except the GUI)
- Inheritance structure for classes is used. Account is a base class and stock account and bank account are derived classes.
- Cash balance is initialize in protected member of account so that it can keep stock account and bank account connected.
- Data structure doubly link list is used to store the stock account portfolio. Node consist of symbol in portfolio and no. of shares user has.
- Also STL map data structure is used to store the data from result1.txt and result2.txt

Here is a brief description of all the files and the functions defined to implement the code–

1. Header files (.h files)

i. account_prakash.h

This is the base class declaration file for the abstract class 'Account'. The pure virtual functions declared here are –

- getbalance ( ) – obtain the cash balance
- setbalance ( ) - set the cash balance
- historyPrint ( ) - Print history of transaction

ii. stock_account_prakash.h

This file has the declaration for the derived class 'StockAccount' which has a public inheritance from the base class 'Account'. The public member functions defined here are as described -

- StockAccount() - constructor
- ~StockAccount() - destructor
- stockDisplay() – to display the stock price
- portfolioDisplay() - to display the current portfolio
- shareBuy() – to buy the shares
- sshareSell() - SELL shares
- listSort() - to sortthe linked list
- portfolioSave() – to save the portfolio
- portfolioGet () - to get the portfolio
- getBalance() – to get the cash balance
- setBalance() – for setting and updating the cash balance
- portfolioValueSave () – for saving the portfolio value and timestamp.
- portfolioValueGet () – to obtain the portfolio value.
- plot_graph() - portfolio value graph plotting
- historyPrint () – to print the stock history

iii. bank_account_prakash.h

- BankAccount() – constructor
- ~BankAccount() - destructor
- getbalance() - virtual function to get cash balance
- setbalance() - virtual function to set cash balance
- balanceDisplay () - view cash balance
- depositCash () - deposit cash
- withdrawCash() - withdraw cash
- historyPrint () – virtual function print history

iv. node_prakash.h

This is the base file for declaration of the doubly linked list nodes with variables for next pointer, previous pointer and related values.

2. Header Function Definition Files (.cpp Files)

i. stock_account_prakash.cpp

This is the file with all the declarations of the functions in the corresponding .h file.

- StockAccount() – constructor to initialise the parameters
- ~StockAccount() – destructor to de-allocate the memory space occupied by the objects
- stockDisplay() – to display the stock price, for user input of the stock symbol, after searching for the stock symbol from the stock database files randomly.
- portfolioDisplay() – to display the current portfolio by accessing the linked list and the container used for storing the values of stock symbol, price per share and the number of shares.
- sharesBuy() – This function facilitates the buying of shares by first checking the user input to be valid in accordance with the requirements of the available shares. It then updates the portfolio and the bank cash balance.
- shareSell() - This function facilitates the selling of shares by first checking if the user input is valid in accordance with the requirements of the shares and then updating the portfolio. It also updates the bank cash balance.
- listSort() – This function implements the sorting of the stock list as per portfolio value. The sorting technique used here is 'Bubble Sort'.
- portfolioSave() - save the portfolio values to a .txt file for future access.
- portfolioGet() - to obtain the portfolio values
- getbalance() – to get the cash balance value
- setbalance() – to set the cash balance value
- portfolioValueSave() – to save the portfolio value and timestamp for further access
- portfolioValueGet() – to get the portfolio value into array for temporary usage
- plot_graph() – plots the variation in portfolio value at MATLAB console
- historyPrint() – to print the stock history of various transactions performed in the lifetime of the code execution

ii. bank_account_prakash.cpp

- BankAccount() – constructor to initialise the parameters

- ~BankAccount() – destructor to de-allocate the memory occupied by the objects of the class.
- getbalance() – to get the cash balance value
- setbalance() - set cash the balance value
- balanceDisplay() – to view the cash balance by accessing the cash balance text file
- depositCash() –to deposit the cash after checking the validity of user input and accessing the text file for balance
- withdrawCash() – to withdraw the cash after checking the validity of user input and accessing the text file for balance
- historyPrint() – to print the history of transactions by accessing text file

3. Main Executable File (.cpp)

main_prakash.cpp

This is the main function file for the entire project. This function employs the switch network for choosing between the various menu options for the Stock Account and the Bank Account.

The first main menu lets user select between the two accounts or exit from the program. The Bank Account menu lets user deposit, withdraw, display balance or get transaction history. The menu for Stock Portfolio lets user buy, sell shares, display portfolio values, display the stock prices, or plot the variation in portfolio values at MATLAB.

4. Text Files (.txt files)

- balance_file - Stores the updated value of cash balance. It is accessed by both the accounts for updated values of cash balance.

- bank_transaction_history – this file stores the list of transaction events along with the time and date. It is updated after every event.

- portfolio_file – This is the main file that stores the entire portfolio information regarding the shares, price and number of shares.

- port_value – this file is a temporary log file for storing information on the variation in portfolio value for plotting the MATLAB graph

- stock_transaction_history - this file stores the list of transaction events along with the time and date. It is updated after every event.

5. Design Patterns

In software engineering, a design pattern is a general reusable solution to a commonly occurring problem in software design. They are of the following types:

- Creational Patterns
- Structural Patterns
  - Adapter, Bridge, Decorator
- Behavioral Patterns
  - Strategy, Template
- Concurrency Patterns

There are two design patterns used in this project –

1. The project makes use of 'template method' design pattern. This method is a type of 'behavioural design pattern' that creates a generic skeleton for information. The template used in this program is of the type 'map'. It is employed as a container that stores two elements – 'key' and its corresponding 'value'.
   The 'StockAccount' class has the definition for this design pattern and it is used in the entire project for iterating through the portfolio values and performing different operations at different stages.

2. Another design pattern used in this project is under the category of 'Creational Design Pattern'. This design pattern is called as 'Singleton' design pattern. In singleton design pattern, the class has only one instance. A global access is set for this design instance.
   In the project, the class 'BankAccount' has a single global instance, thus implementing singleton design pattern.