National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Course Project – Programming Fundamentals [Fall 2022]

Cybersecurity Department

Deadline: Wednesday, 07-12-2022

You have to make a simulator of stock market trading application which runs as follows:

Load/save Stock Market Data: Read companies.txt file and store data in a 2-D array. The file contains comma separated three values in each line – stock symbol, company name, current price. When the application is closed, i.e. session ends, new prices of each stock will be saved such that companies.txt will have updated stock prices.

C++ concepts to use	Read from file, using 2-D array

Show Stock Market Screen: Once all data is loaded, live stock market screen is displayed as shown in the figure below. Each shown stock will have its symbol, company name, offer price which is used to sell or buy this company share, high of current session i.e. only for this run of the program, low of this session. All this data except high and low are loaded from the file and once the market session is live (with the beginning of the program), the offer price will fluctuate. On every press of ENTER key, following actions are performed (see starter code):

- all stocks values are changed randomly. Stock price cannot get an increase or decrease more than 15% in one run of the program. For example, if a share is Rs. 100 today, it cannot go beyond Rs.115 or less than Rs.85. It can however be 115 and we close the program. This 115 will be stored in the file which means that next time program will load 115 value. And then 15% increase/decrease cap will be applied on 115. When the program finds the random increase or decrease, it should be in both integer and floating part, that is, 150.70 is changed to 151.81 by a random increase of 1 in the integer part and .11 increase in the floating part. Use parallel arrays to keep symbols of the stocks and their current/live prices.
- Keep track of all shares bought/sold in the current session (this functionality is explained later)
- Calculate percentage of price increase and decrease of each share and store it (possibly in a parallel array). Then you will be required to show top advancer and decliner.
- Refresh/redraw live stock market screen. In linux systems, you can use system("clear") function to clear all screen text and getch() of conio.h library to get next user input. If user presses ENTER key, values are changed and displayed again; for other keys, appropriate action is performed. Remember that when you clear screen, everything will be cleared therefore you will need to redraw the Karachi Stock Market Live screen. Use a function which redraws with new values.

C++ concepts to use	iomanip, functions, loops, arrays, strings, random numbers, explore library function to convert string to float Up arrow and down arrow can be printed using their Unicode (extended version of ASCII) characters. Try printing "\u2191" and "\u2193".

Karachi Stock Market (Live)					
Show updates: ⏎ Show Portfolio: P Add Stock: A Remove Stock: R Add money to account: M Exit: E					
Stocks	**Company Name**	**Prev price**	Curr Price	High	Low
OGDC	Oil and Gas Development Company Limited	150.23	**151.81 ↑**	154.20	149.70
PPL	Pakistan Petroleum Limited	480.5	**485.75 ↑**	493.40	485.00
POL	Pakistan Oilfields Limited	72.6	**69.76 ↓**	73.9	69.60
MARI	Mari Petroleum Company Limited	77	**76.81 ↓**	78.65	76.13
...					
....	All companies listed as per companies.txt				
Total shares traded today	2890045				
Top % advancer symbol	TRG				
Top % decliner symbol	HBL				

User Portfolio: User portfolio is the collection of all shares of different companies that user buys. Moreover, it will show user account balance and any profit/loss. Initially, this list will be empty and user will not have any balance. When portfolio screen is active, user will have choice to enter his/her name, add stock to portfolio, remove stock from portfolio, add money to account, withdraw money from account, show Live Market screen.

Portfolio owner: Bill Gates (Live)							
Show updates: ⏎ Live Market: L Add Stock: A Remove Stock: R Add money to account: M Withdraw money: W							
Stocks	**Company Name**	shares	Current	Previous	Gain/Loss	High	Low
PSO	Pakistan State Oil	1000	151.81	150.70	+1110	154.20	150.70
LUCK	Lucky Cement	500	485.75	488.20	-1225	493.40	485.00
HBL	Habib Bank Limited	2500	69.76	70	-120	70	69.60
OGDC	Oil & Gas Dev. Comp.	3000	76.81	76.40	+1230	78.65	76.13
Today's Gain or Loss (Rs.)					+995		
Previous Balance (Rs.)					+78956		
New Balance (Rs.)					+79951		

Load/Store user Portfolio: When program is run for the first time, there won't be any portfolio.txt file. After first run, user will enter portfolio data in the portfolio screen which will be displayed and also stored in portfolio.txt file when program ends.

C++ concepts to use	Read from file, Write to file (iomanip for bonus marks)

Show Portfolio Screen: Data of portfolio screen can be figured out from the portfolio screen shown in the figure above. This will also work like main market screen, that is, user presses ENTER to get new values of stocks and that should show changes in the table. Make sure that when you update stock values randomly, its done exactly the same way as you did in the market screen, and for ALL stocks (not only for portfolio stocks).

C++ concepts to use	iomanip, functions, loops, arrays (search), strings

ADD Stock to Portfolio: Keep a list of stock symbols for the user. Add new stock in the list when user chooses to add new stock. Adding a stock means that user buys shares of that stock. So the program should ask the number of shares to buy and for which stock. Suppose user buys 10 shares of Meezan Bank and current price of its share is 113 then Rs. 113 will be deducted from the user balance and Meezan symbol will appear in the user portfolio. Make sure that user that enough balance to buy the shares. Show appropriate errors if balance is not enough. Also make sure that user enters correct

symbol, for example, if user enters PZO to buy instead of PSO then program should ask user to enter symbol from the list.

C++ concepts to use	functions, loops, arrays (search), strings

REMOVE stock from Portfolio: When user choses to remove stock, he/she enters the symbol and number of shares to sell. Balance will increase by the amount user gets on selling the stock on the current price. Do input validations.

C++ concepts to use	functions, loops, arrays (search), strings

ADD/Withdraw Money: User will be able to add or withdraw money from the account. Do validations such as trying to withdraw money more than balance etc.

C++ concepts to use	functions, loops, arrays (search)

Rubric:

Name	
Registration#	
Class Section	

Task Implementation	**Marks**	**Obtained**
Load stock market data	10	
Show Stock Market Screen	10	
In market screen: calculation of **random increase/decrease**, **high/low**, **total shares traded**, **top advancers/decliners**. [4 x 5 marks]	20	
Load and Store portfolio data in portfolio.txt file (5 + 5 marks)	10	
Show portfolio screen (almost same as Stock Market Screen Live)	10	
In Portfolio screen: **calculation of gain/loss**, **keeping record of high/low**, **total gain/loss**, **current/previous balance**	20	
Add stock to portfolio (use colours for bonus marks)	10	
Remove stock from portfolio	10	
Add/Remove money from account	10	
TOTAL MARKS	**110**	
Bonus marks		
Use text and background colours similar to the output in the figure. Gain will be green and loss will be indicated by red.	10	
Save updated market data on closing the application (closing session)	10	
Use dynamic array to keep list of user stock	10	
portfolio.txt is properly formatted using iomaip	10	
Deduction scenarios		
Not properly indented/commented code.	-10 marks	
Your project cpp file should have your Name, Roll number, Class section in comments at the top.	-10 marks	
Illogical variable names, wrong type of variables used	-10	
Global variable used	-100%	
Using constructs like GOTO, OOP, etc not taught in the course	-100%	
Plagiarism	-100%	
Cannot redo task given in demo (could write similar code but is incorrect)	-50%	
Cannot redo task given in demo (totally blank, clueless)	-100%	

Critical Restrictions/Instructions

- If global variable is used, your project will not be checked and you will get ZERO.
- If a programming construct other than those taught in the course is used, your project will get ZERO.
- If plagiarism is found, your project will get ZERO.
- During viva you will be asked to rewrite some portion of code, do a small task again that you have done in the project. In case of not been able to do that your project will not get more than 50% marks in any case even if your project is excellent. This is to make sure that you do it yourself. However, if evaluator finds that you are totally blank and directionless, it will be deemed that you have not done it yourself and you will get ZERO.
- You are not supposed to get any help from Internet or peers in making your project logic. Internet will have similar projects so looking at them will result in plagiarism.

Submission Guideline

- Submission before deadline. No late submission will be accepted. Keep your work on google drive. No excuse of computer crashed etc will be accepted.
- Make sure that you submit correct files by downloading and checking them after your submit. No excuse of mistakenly wrong file uploaded will be accepted.
- You must submit self-marked rubrik along with your code on due date and bring a printout of the same for the viva.
- Demo/viva dates will be announced. Viva will be of 15-20 minutes where you will be asked questions from your submitted code, you will run your program, and you could be given a task to perform.
- You will get ZERO in the project if you miss the viva. In case of unavoidable circumstances, you must submit a valid documentary proof and reschedule your viva BEFORE your allotted time.
- **File naming and submission**:We use auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.
    - Run and test your program on a lab machine before submission.
    - Your project file should be named as ROLL-NUM_SECTION_Project.cpp (e.g. **22i-0001_A_Project.cpp**).
    - Combine all your work (all files including companies.txt, self marked rubric, etc) in one folder. The folder not have any binaries, no exe files etc. If we unable to download your submission due to any reason you will be awarded zero mark.
    - Rename the folder as ROLL-NUM_SECTION (e.g. 22i-0001_A) and compress the folder as a zip file. (e.g. **22i-0001_A.zip**). Only zip file will be acceptable.
    - Submit the .zip file on Google Classroom within the deadline. Make sure that you have submitted the correct file.
    - Submission other than Google classroom (e.g. email etc.) will not be accepted.
- The student is solely responsible to check the final zip files for issues like corrupt files, viruses in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.