CS 2401 – Spring 2022

Project 1 – Checkbook Class

Due: 11:59 PM Friday, February 4th

A checkbook is a container that holds a record of all the checks written by the owner. Our container, being computerized, will offer the user a variety of extra features so that they can find, organize, and better understand their check-based spending.

Begin by copying into your working directory the files that are already written for you for this project. They are

- date.h
- date.cc
- check.h
- check.cc
- commented_main.cc
- bob_checks.txt

The check.h and check.cc files define a class called Check, which holds the information for a single check, including the check_number, the date that it was written, to whom it was written and the amount of the check. This class is dependent on the Date class which I have provided. You will **not** be changing anything about the Date class, nor anything about the check.h file, but you will need to complete check.cc. Note that the Date class already has defined the >> and << operators for Date objects as well as a full set of comparison operators. You will be using these. Do not try to re-invent nor circumvent these operators. (e.g. Users will type in dates as `9/6/2017` – not as separate numbers from which you reassemble a date.)

In the check.cc class you will need to write the implementations for the write_check function (an input function) and the output function.

Next you are to create checkbook.h and checkbook.cc. Your checkbook objects will consist of an array capable of holding 200 checks, a number to keep track of how full that array is, a balance (which says how much money is in the checking account), ***and a variable for the next available check num.*** By looking at the main, you can see the names of the functions that you are to write for the checkbook class. (The main that I gave you is also a file you should not change except to uncomment the checkbook class calls.) These will give the user the ability to:

1. Have their checkbook reloaded from the backup file – so they do not re-enter their checks every time they start the program.
2. Make a deposit into their checkbook.
3. Write a check – the user **can*not*** enter the check-number, ***the check number will be inserted by the checkbook using the set_chk_num() function of the Check class***.
4. See the checkbook balance.

5. See a listing of all the checks that they have written – for each check they should see all the information about that check.
6. Remove a check by entering its check-number. ***Once a check is removed that check number is never available again.***
7. Sort the checks by check-number.
8. Sort the checks alphabetically by the person to whom they were written.
9. Sort the checks by the date they were written.
10. Find and view all the checks written to a particular payee along with the total amount of those checks.
11. Find the average of all the checks written.
12. Have the checkbook backed up to the same file that it was read from at the beginning of the program, upon exiting the program (the main calls the save function – you write the save function)

Note that items 2 – 11 above are menu options that are done in a loop in the main. The back-up file is done without the user's knowledge or interaction.

Each file that you create should have a header block with your name, an approximate date, and a description of what is done in the file. Also, your container class should exhibit a correct use of const and static const.

Submit check.cc, checkbook.h, checkbook.cc, and your own data file on Blackboard.