CS 2401 HW2, Auto Parts Store
50 Points

Objectives

An auto part store wants to keep track of the auto parts they sell and their sale transactions. Each part has the following properties:

- Name/description
- Number/ID
- Price
- Quantity on hand
- List of transactions

Design a C++ class called **Part** that keeps track of the part information and its transactions. Include all the appropriate member functions for the part (constructors, getters, setters, etc.).

The list of transactions must be implemented using a **dynamic array**. Once your program starts, dynamically allocate an array of size 5 (default). **Do not use any STL library to keep track of the transactions.**

Each transaction includes the following:

- Transaction ID
- Transaction date (format: "2021-07-30")
- Quantity

Write a C++ program that reads all the parts' information from a file followed by its transactions and stores everything in a Part object. The file is formatted as follows:

```
Part Name (may contain spaces)
Number   Price   Quantity
Transaction ID    Date    Quantity
Transaction ID    Date    Quantity
Transaction ID    Date    Quantity
Transaction ID    Date    Quantity
….
```

CS 2401 HW2, Auto Parts Store
50 Points

Your program should display a menu of options as follows:

1. Print the part and all of its transactions in a nice format. This must be done in a separate function that is part of your main program (**not a member or a friend function**).
2. Add transaction. The transaction should be rejected if the quantity exceeds the quantity on hand. A transaction ID starts at 1000 and gets incremented automatically based on the last transaction recorded
3. Search for a transaction by date
4. Search for transaction by id
5. Delete all transactions (ID starts at 1000 again)
6. Quit and save all transactions. Save everything back to the same input file.

Your program should read the current date when it starts in the format: "2021-07-30". Your program should accept any number of transactions (i.e., the list should grow dynamically). You will be graded on your design as well as the functionality of your program. Follow the guidelines outlined in the lecture videos concerning classes with dynamic allocation (**Big Three**).

Create a Makefile to compile your program. See the Makefile sample given in the last assignment.

Grading:

Programs that contain syntax errors will earn zero points.
Programs that use global variables, other than constants, will earn zero points.
Programs that use STL libraries such as vector will earn zero points.

45 Points:

- (15 points) Class design
- (25 points) Program is functioning as described above with the five options
- (5 points) Results are displayed in a nice format

5 points: Programming Style & documentation.

- All files must have your name, date, and a description of their content
- All functions must be documented in the header file using the style outlined in the coding style below

Follow the coding style outline on GitHub:
https://github.com/nasseef/cs/blob/master/docs/coding-style.md

Submission:

Submit a link to your repository on Blackboard before the deadline.