

## Project 1 – Container Class

Due: 11:59 PM Friday, February 3<sup>rd</sup>

You have been asked by a plant nursery to create an inventory system for their business. To do so, you will need to write a container class that holds records describing each plant and how many are in stock.

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

- date.h
- date.cc
- plant.h
- plant.cc
- commented\_main.cc
- inventory.txt

The plant.h and plant.cc files define a class called Plant, which holds the information for a single plant, including the name, the color, the date these plants came in stock, and how many are in stock. 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 plant.h file, but you will need to complete plant.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 8/30/2022 – not as separate numbers from which you reassemble a date.)

In the plant.cc class you will need to write the implementations for the input and output functions.

Next you are to create plantNursery.h and plantNursery.cc. Your PlantNursery objects will consist of an array capable of holding 200 plants and a number to keep track of how full that array is. By looking at the main, you can see the names of the functions that you are to write for the PlantNursery class. (The main that I gave you is also a file you should not change except to uncomment the PlantNursery class calls.) These will give the user the ability to:

1. Have their inventory list reloaded from the backup file – so they do not have to manually re-enter their plants every time they start the program.
2. See a list of all the plants in stock
3. Add a new type of plant that just came into stock.
4. Remove a plant that is sold out.
5. Change the amount of a plant that is currently in stock.
6. Sort the plants alphabetically by name.
7. Sort the plants by the date they came in stock.
8. Sort the plants by the quantity in stock.
9. Find and view all the plants of a certain color (with the total).

10. Find and view all the plants that came in stock before a certain date.
11. Find and view all the colors of a certain plant type (i.e. plants with the same name).
12. Find the average number of plants that you keep in stock (i.e. the average of the “stock” attributes).
13. Have the PlantNursery data 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)

Notes:

1. Items 2 – 12 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.
2. Options 6 – 8 should each use a different sorting algorithm.
3. Remember to declare the size of the array as a static constant in the public section.
4. 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.
5. Any header files should include Doxygen style comments for each function and you should add descriptive comments throughout your code.
6. Your container class should exhibit the correct use of const and static const.
7. You can read in and output dates like this:

```
Date d;  
cin >> d;  
cout << d;
```

Don’t waste time rewriting that code!

8. When you read in a string, it is best to use getline so you make sure to get the entire string (including spaces). Also remember to ignore any leftover newline characters when using getline after the >> operator.
9. You must pass streams by reference.
10. For each plant in stock, the format of the data file is:  
plant\_name  
plant\_color  
came\_in\_date  
stock\_amount

Submit plant.cc, plantNursery.h, plantNursery.cc, and your own data file on Blackboard.