

Project Three: Corner Grocer Item Frequency

Tracker

Kenya Crow

CS 210 – Programming Languages

Instructor Bill Chan

10/17/2025

Overview

This project involved creating a C++ application for a fictional store called the Corner Grocer. The goal of the program is to analyze which grocery items were purchased throughout the day, based on a straightforward text file of purchase records. The program helps the store figure out which items are most frequently bought, so they can make better decisions about where to place products for maximum convenience and visibility.

The program reads from a file of item names (one item per line), counts how many times each item appears, and gives the user a few options to interact with that data.

What the Program Does

When the program starts, it immediately reads the input file called CS210_Project_Three_Input_File.txt, and stores how many times each item shows up. It uses a `std::map` to efficiently keep track of item names and their counts.

It also automatically creates a backup file called `frequency.dat`, which saves each item along with its frequency; so, there is always a copy of the data even if the program closes.

After that, the user can interact with the program through a menu with four options:

1. Search for an item – Type in an item name, and the program will tell you how many times it was purchased that day.
2. List all item frequencies – Print out every item and how many times it was bought.

3. Show a histogram – Display a visual chart using asterisks, where each asterisk represents one purchase.
4. Exit – Close the program.

All input is validated to make sure the user picks a valid menu option.

How It's Built

The code is organized around a class called `GroceryTracker`, which handles all the logic for loading data, tracking item counts, printing results, and saving the backup file.

Here is a quick breakdown of the important parts:

- Item Frequency Tracking:

Uses a `std::map<std::string, int>` to store items and how often they were bought.

- File Handling:
 - Reads the input file during startup
 - Creates `frequency.dat` automatically, without needing the user to do anything

- Menu System:

A basic text-based menu allows users to interact with data in diverse ways.

- Clean Code:

The code follows standard practices; clear variable names, comments to explain logic, and organized structure using functions and classes.

Screenshots

```
main.cpp CS210_Project_Th... frequency.dat
1 Apples 4
2 Beets 3
3 Broccoli 7
4 Cantaloupe 2
5 Cauliflower 6
6 Celery 6
7 Cranberries 10
8 Cucumbers 9
9 Garlic 8
10 Limes 1
11 Onions 4
12 Peaches 5
13 Pears 1
14 Peas 8
15 Potatoes 5
16 Pumpkins 2
17 Radishes 3
18 Spinach 5
19 Yams 5
20 Zucchini 10
21

3. Display histogram of item frequencies
4. Exit
Enter your choice (1-4): 3

Item Frequency Histogram:
-----
Apples      ****
Beets       ***
Broccoli     *****
Cantaloupe   **
Cauliflower  *****
Celery       *****
Cranberries  *****
Cucumbers    *****
Garlic       *****
Limes       *
Onions       ****
Peaches      *****
Pears        *
Peas         *****
Potatoes     *****
Pumpkins     **
Radishes     ***
Spinach      *****
Yams         *****

===== Corner Grocer Menu =====
1. Search for item frequency
2. Display all item frequencies
3. Display histogram of item frequencies
4. Exit
Enter your choice (1-4): 2

Item Frequencies:
-----
Apples      4
Beets       3
Broccoli     7
Cantaloupe   2
Cauliflower  6
Celery       6
Cranberries  10
Cucumbers    9
Garlic       8
Limes       1
Onions       4
Peaches      5
Pears        1
Peas         8
Potatoes     5
Pumpkins     2
```

What I Learned

This project was a good example of putting together multiple programming concepts — file I/O, classes, maps, and menus — into a working application. It also highlighted the importance of designing programs that are user-friendly and maintainable.

If I were to improve it further, I would consider:

- Making item searches case-insensitive
- Saving the histogram output to a file
- Allowing the user to input data directly (not just from a file)

References

None used. All logic and structure were based on course materials and personal understanding of C++.