A logo of a grocery bag

AI-generated content may be incorrect.

**Corner Grocer Project**

**Developed by: Shelby Joiner**

**Role: Junior Developer @ Chada Tech**

**Project 3: Item Frequency Tracker**

**Project Overview**

**In this assignment, I was tasked with building an item frequency tracking program for the Corner Grocer. As a junior developer at Chada Tech, I wanted to deliver a solution that not only met the specifications but went beyond expectations by using clean, efficient, and well-documented C++ code.**

**This program reads through a text-based record of daily grocery transactions and tracks the frequency of each item purchased. The store plans to reorganize their produce section based on these statistics, so accuracy and usability were top priorities.**

The program is called "Corner Grocer." It helps to track how often items are purchased by reading data from a text file. The user can interact with a menu to:

Search for the frequency of a specific item (case-insensitive),

Display the frequency of all items,

Display a histogram (a visual bar chart) of item purchase counts,

Exit the program.

Libraries & Includes

#include <iostream>

#include <fstream>

#include <map>

Purpose: These lines load in libraries (also known as "includes") that the program will use:

iostream helps us display messages and get input from the user (e.g., printing to the screen or reading input).

fstream is used for reading data from files (in this case, the input file that contains item data).

map helps us store and organize items along with their purchase counts.

Helper Function: toLowercase

string toLowercase(string str) {

for (char& c : str) {

if (c >= 'A' && c <= 'Z') {

c += 32; // Convert uppercase to lowercase

}

}

return str;

}

Purpose: This helper function converts any given string (like item names) to lowercase. This is helpful because it allows the program to compare item names without worrying about whether the user types "apple" or "Apple."

How it works: It looks at each character in the string, checks if it’s an uppercase letter, and if so, converts it to a lowercase letter.

Main Program Logic

1. Setting up Data Structures and File Input

cpp

Copy

Edit

map<string, int> itemFrequency;

map<string, string> lowercaseToOriginal;

string fileName = "CS210\_Project\_Three\_Input\_File.txt";

ifstream inputFile(fileName);

Purpose:

itemFrequency: This stores the items (with original capitalization) and their frequency of appearance.

lowercaseToOriginal: This stores the lowercase version of an item as the key, and the original item name as the value. This allows the program to store items in their original form while still performing case-insensitive searches.

fileName: Specifies the name of the file that contains the list of purchased items.

ifstream inputFile: Opens the file to read the item data.

2. Error Handling for File Opening

if (!inputFile) {

cerr << "Error: Could not open " << fileName << endl;

return 1;

}

Purpose: This block checks if the program is able to open the file. If it can't (for example, if the file is missing or there's an issue with the file path), it displays an error message and exits the program.

3. Reading and Counting Item Frequencies

string item;

while (inputFile >> item) {

string itemLower = toLowercase(item);

if (itemFrequency.find(item) == itemFrequency.end()) {

lowercaseToOriginal[itemLower] = item;

}

itemFrequency[lowercaseToOriginal[itemLower]]++;

}

inputFile.close();

Purpose:

This loop reads the item names one by one from the input file.

It converts the item name to lowercase for case-insensitive lookup and stores it.

The program then checks if the item is already in the itemFrequency map:

If it's the first time encountering this item, the program stores the original case version in lowercaseToOriginal for future reference.

It then increments the count for that item.

Why it’s useful: This is how the program tracks how many times each item appears in the file.

4. Main Menu Loop

int choice;

while (true) {

cout << "\n==== ItemFrequencyTracker Menu ====\n"

<< "1. Search for Item Frequency\n"

<< "2. Display All Item Frequencies\n"

<< "3. Display Histogram\n"

<< "4. Exit\n"

<< "\nChoose an option (1-4): ";

cin >> choice;

Purpose: This starts a continuous loop that shows the user a menu with four options to interact with the program:

Search for the frequency of an item.

View all item frequencies.

See a visual histogram of item frequencies.

Exit the program.

How it works: The program waits for the user to input a number (1-4) to choose an option.

5. Input Validation

if (cin.fail()) {

cin.clear();

cin.ignore(1000, '\n');

cout << "Invalid input. Please enter a number 1-4.\n";

continue;

}

Purpose: This ensures the user inputs a valid option (a number between 1 and 4). If the user types something invalid (like a letter), it displays an error message and prompts them again.

6. Option 1: Search for Item Frequency

if (choice == 1) {

cout << "Enter item name: ";

string searchItem;

cin >> searchItem;

string searchLower = toLowercase(searchItem);

if (lowercaseToOriginal.find(searchLower) != lowercaseToOriginal.end()) {

string actualItem = lowercaseToOriginal[searchLower];

cout << actualItem << " appears " << itemFrequency[actualItem] << " time(s).\n";

}

else {

cout << searchItem << " was not found in the records.\n";

}

}

Purpose: This option allows the user to search for the frequency of a specific item.

The user types an item name (e.g., "Apple"), and the program converts it to lowercase.

It then checks if the item exists in the list (ignoring case) and displays how many times the item appears. If the item isn’t found, it lets the user know.

7. Option 2: Display All Item Frequencies

else if (choice == 2) {

cout << "\nItem Frequencies:\n";

for (const auto& pair : itemFrequency) {

cout << pair.first << " " << pair.second << endl;

}

}

Purpose: This option displays all the items and how many times each one appears.

How it works: It goes through the itemFrequency map and prints each item along with its frequency.

8. Option 3: Display Histogram

else if (choice == 3) {

cout << "\nPurchase Histogram:\n";

for (const auto& pair : itemFrequency) {

cout << pair.first << " ";

for (int i = 0; i < pair.second; ++i) {

cout << "\*";

}

cout << endl;

}

}

Purpose: This option shows a simple graphical representation (a histogram) of item frequencies.

How it works: It prints out each item followed by a number of stars (\*) corresponding to the frequency of that item. For example, if "Apple" appeared 3 times, it would display "Apple \*\*\*".

9. Option 4: Exit

else if (choice == 4) {

cout << "Exiting program. Goodbye!\n";

break;

}

Purpose: This option allows the user to exit the program. When selected, the program prints a goodbye message and exits the loop, effectively ending the program.

10. Invalid Option Handling

else {

cout << "Invalid option. Please choose 1-4.\n";

}

Purpose: If the user inputs anything other than 1-4, the program lets them know that the input is invalid and asks them to try again.

Final Return Statement

return 0;

Purpose: This marks the successful completion of the program. It tells the operating system that the program has finished running without errors.

Summary

This program is a useful tool for tracking and managing the frequency of grocery items. It reads data from a file, allows users to search for item frequencies, view all items, see a visual representation of item counts, and exit the program when done. It’s a simple yet effective way to manage and analyze purchase data.  
  
Final Screenshot with validation testing:   
A screenshot of a computer program

AI-generated content may be incorrect.