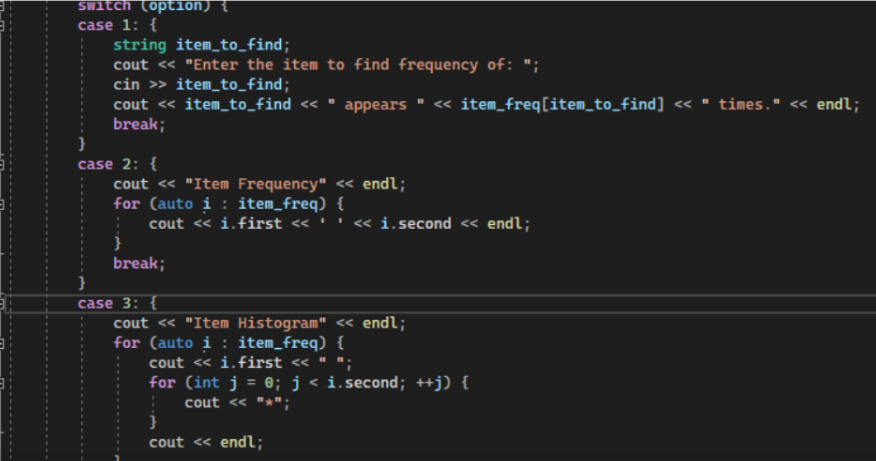
Anthony Shewmake

12/15/2024

Summary of project 3

This project is a command-line program that reads in a file of grocery items and their frequencies and allows the user to search for a specific item, print a frequency list, or print a histogram of the frequencies using asterisks

The program reads in the input file, constructs a vector of Grocery objects, and performs various operations on that vector based on user input. The program solves the problem of analyzing and visualizing data from a grocery list.



This code is a program that reads in a file containing a list of groceries, constructs a list of Grocery objects from that file, and then presents a menu that allows the user to search for a specific item, print a frequency list of all the items in the file, or print a histogram of the items in the file using "" symbols. The Grocery class has a constructor and accessor/mutator methods for the item name and quantity. The main function uses these methods to manipulate the list of Grocery objects. The backup Data function writes all the Grocery objects to a file named "frequency.dat". The read Groceries From File function reads in the input file and constructs the list of Grocery objects using a map to count the number of times each item appears in the file. The program presents a menu with options to the user. Option 1 allows the user to search for a specific item by name and returns the number of times that item appears in the file. Option 2 prints a frequency list of all the items in the file, showing the item name and the number of times it appears in the file. Option 3 prints a histogram of the items in the file using "symbols", where each item is represented by its name and the number of "\*" symbols corresponding to the number of times that item appears in the file. Option 4 exits the program.

One possible area for improvement is error handling. The program currently outputs error messages but does not necessarily exit or handle the error in a graceful way. Improvements such as adding more specific error messages or implementing try-catch blocks could make the code more secure and user-friendly.

The most challenging piece of the code was constructing the histogram using asterisks. However, the use of the startHelper() function to generate a string of asterisks based on the frequency value made the code more efficient and readable.

The skills from this project that will be transferable to other projects include working with vectors, maps, and file input/output; implementing search algorithms; and creating formatted output. Additionally, the project reinforces the importance of code organization and commenting for readability and maintainability.

To make the program maintainable, readable, and adaptable, the code is organized into appropriate functions, uses clear and descriptive variable and function names, and includes comments where necessary. The code is also designed to be easily modifiable by adding new functions or modifying existing ones.