CS 210 – Project Three: Corner Grocer Item Tracking Program

Name: Maximo Obra Winfield

Date: 2/18/2025

The Corner Grocer Item Tracking Program is designed to analyze and track grocery items purchased throughout the day. The program read an input file containing a list of purchased items, counts the frequency of each item, and provides various output options for the user. The program also creates a backup file (frequency.dat) to store item frequency data for future reference.

The program is structured using an ItemTracker class, which encapsulates the logic for reading files, counting item frequencies, storing data in a std::map, and providing functionality to search, display, and export frequency data. The main menu allows users to interact with the program by selecting one of four options:

1. Search for an Item’s Frequency
   * The user inputs an item name, and the program returns the number of times it appears in the data.
2. Print the Full Frequency List
   * The program prints a list of all items along with their purchase counts.
3. Print a Histogram
   * The program prints a text-based histogram where the number of asterisks (\*) represents the frequency of each item
4. Exit the Program
   * The user can choose to terminate the program

The program also performs input validation to prevent errors from invalid entries.

**Code Implementation**

* **File Handling:** The program reads from CS210\_Project\_Three\_Input\_File.txt and creates a backup file frequency.dat
* ****Data Structures:** A std::map<string, int> is used to store item frequencies.**
* ****Encapsulation:** The ItemTracker class ensures that file reading, data processing, and output generation are handled separately from the main function**
* ****Memory Management:** The program ensures all file streams are properly opened and closed.**

**Performance Considerations**

The use of std::map for frequency tracking provides efficient operations:

* Insertion and retrieval operations have an average time complexity of O(log n) due to the balanced tree structure of std::map.
* Search operations are also O(log n), ensuring quick lookups even with large datasets.
* Compared to an unordered\_map (O(1) lookups but unordered storage), std::map maintains items in sorted order, which is beneficial when displaying frequencies in order.

This design ensures that the program remains **efficient and scalable**, even when processing large datasets of grocery items.

**References**

**1. Stroustrup, B. (2013). The C++ programming language (4th ed.). Addison-Wesley.**

* **This is a fundamental reference for C++ programming concepts, including file handling and map usage**

**2. Standard C++ Foundation. (2024). C++ reference: Map (std::map). Retrieved from** <https://en.cppreference.com/w/cpp/container/map>

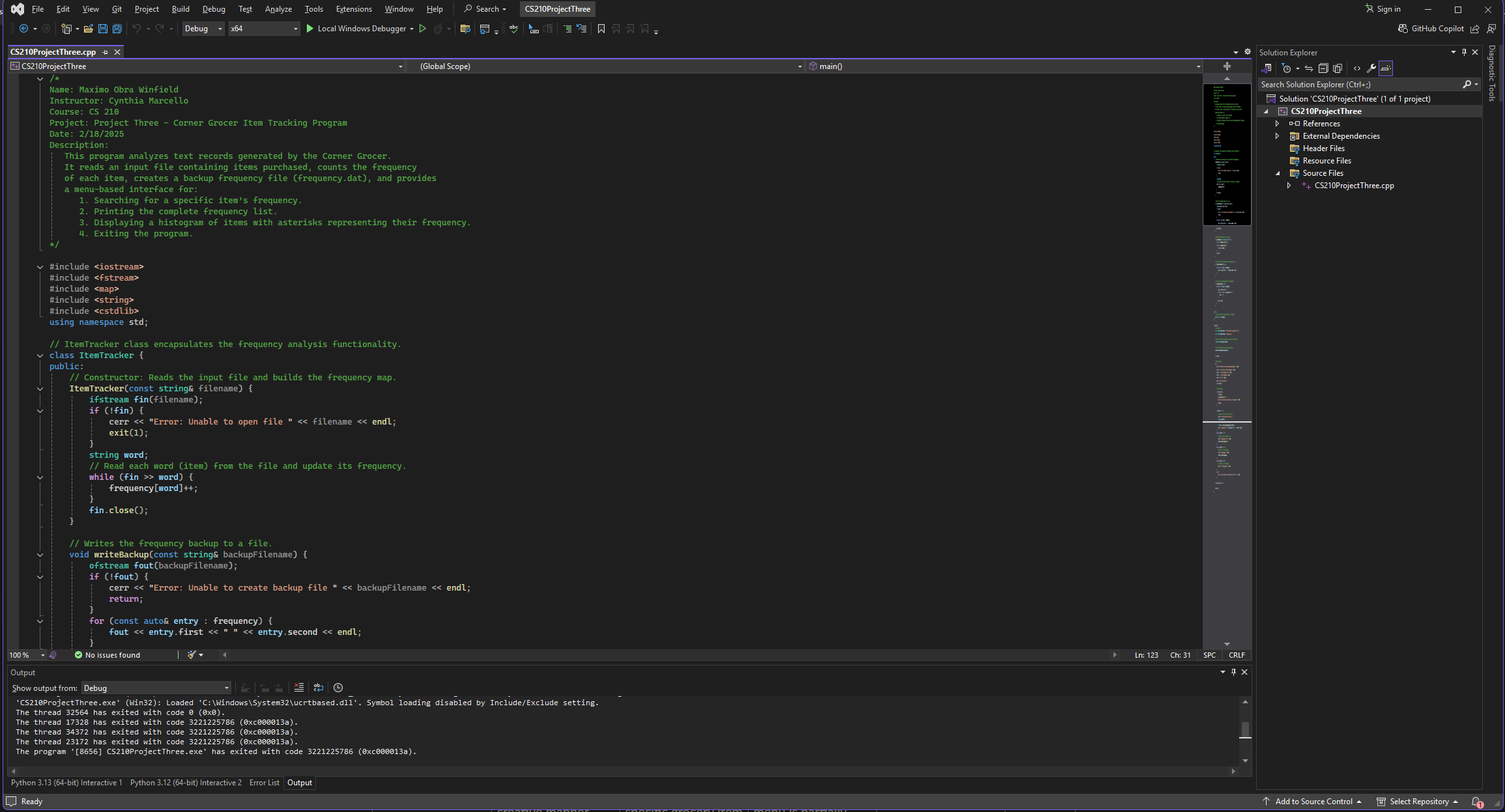
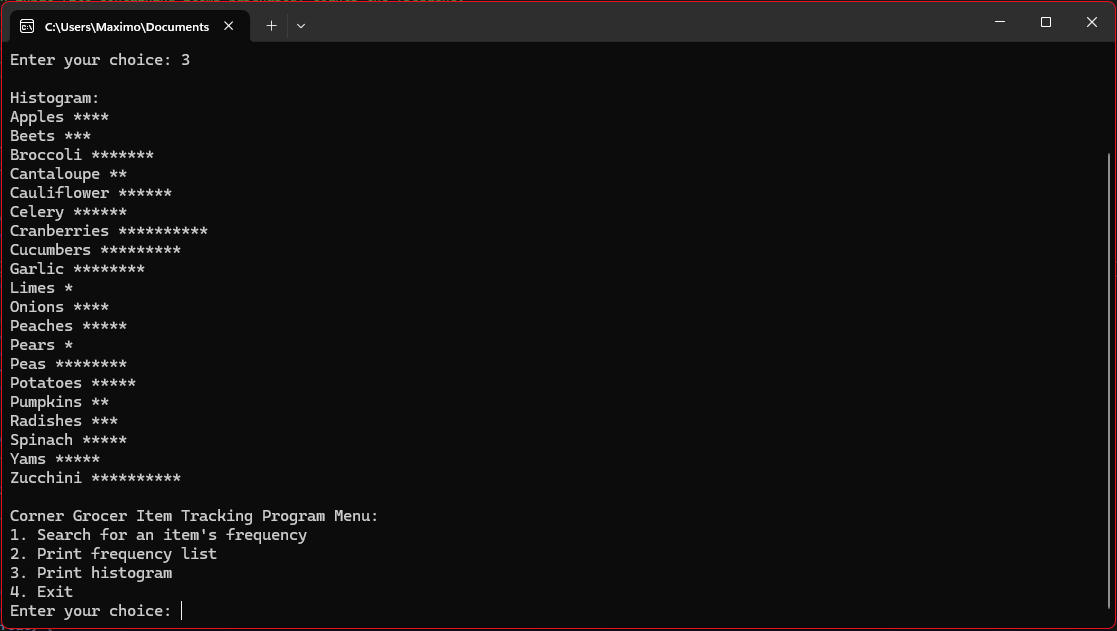
3. Microsoft. (2024). File handling in C++ (ifstream, ofstream, fstream). Retrieved from <https://learn.microsoft.com/en-us/cpp>

* This source explains file operations in C++ using ifstream and ofstream

4. Gaddis, T. (2020). Starting out with C++: From control structures through objects (10th ed.). Pearson.

* This book provides an introduction to input validation, class design, and file operations in C++

**Screenshots**

Source Code in Visual Studio (demonstrates program structure)Histogram Output in Console (shows correctly formatted output)