***Display*** manages the user interface for a program interacting with data.

Constructor (Display()): Initializes output properties and sets a default header symbol.

Main Menu (void MainMenu()): Displays the main menu of the program. The current options are search item, item quantity, item quantity graphed, and quit. These options can be updated in an array for maintainability.

List All Items (void ListAllItems(map<string, int> parsedData)): Displays all items and their frequencies from a data map.

Item Search Screen (void ItemSearchScreen()): Displays a prompt for searching an item. Additional option so list all the items.

Item Frequency Number Screen (void ItemFrequencyNumberScreen(map<string, int> parsedData)): Displays item frequencies as numbers.

Item Frequency Asterisk Screen (void ItemFrequencyAsteriskScreen(map<string, int> parsedData)): Displays item frequencies as a histogram of '\*' characters.

Header Template (string HeaderTemplate(string title)): Formats a header string with a centered title using a specified header symbol.

Print N Times (string PrintNTimes(size\_t numTimes, char repeatSymbol)): Generates a string consisting of a character repeated n times.

Get Header Symbol (char GetHeaderSymbol()): Retrieves the header symbol.

Set Header Symbol (void SetHeaderSymbol(char newChar)): Sets a new char as the header symbol.

***DataInterpreter*** processes raw data into parsed data with item frequencies.

Constructor (DataInterpreter(vector<string> inputData)): Initializes with raw data and calculates item frequencies.

Display Raw Data (void DisplayRawData()): Prints all raw data items.

Get Parsed Data (map<string, int> GetParsedData()): Returns parsed data as a map with the “Key” being the individual strings and the “Values” being the count of those items.

Item Search (void ItemSearch(string itemToSearch)): Searches for and displays the name and quantity of a specific item.

Data Calculator (void DataCalculator()): Calculates frequencies of items in raw data and stores them in a map.

***WriteFile*** is responsible for writing data from a map to a specified file.

Constructor (FileWriter(string filePath, map<string, int> mapData)): Initializes with file path and data map, writes data to file.

Write File Data (void WriteFileData()): Writes the items and their frequencies to the file.

***FileReader*** is responsible for reading data from a file to a vector.

Constructor (FileReader(string filePath)): Initializes with a file path.

Read File Data (void ReadFileData()): Read the items from the file into a vector.