**Code Design and Functionality Documentation**

This provides a comprehensive description of the design and functionality of the item-tracking program developed for Corner Grocer. The program reads a text file containing items purchased throughout the day, counts the frequency of each item, backs up this data to a file named *frequency.dat*, and offers an interactive menu for users to query and visualize the data.

***Design and Structure***  
The solution is written in C++ and leverages the Standard Template Library (STL) map to store item frequencies. The code is modular and organized into several functions:

* **printMenu()**: Displays the menu options to the user.
* **getValidatedChoice()**: Implements input validation by ensuring that the user’s menu selection is an integer between 1 and 4. This function uses a loop to clear erroneous input and reprompt the user when needed.
* **writeFrequencyData()**: Writes the frequency count of each item to the backup file (*frequency.dat*), ensuring that the data can be restored or audited later.

***Functionality***  
Upon execution, the program performs the following steps:

1. Opens and reads *CS210\_Project\_Three\_Input\_File.txt*, incrementing the count for each word encountered.
2. Backs up the accumulated frequency data into *frequency.dat*.
3. Enters a menu loop that provides:
   * A search feature to display the frequency of a specific item.
   * A complete list of items with their respective frequencies.
   * A text-based histogram where each item is followed by a number of asterisks corresponding to its count.
   * A graceful exit option (Option 4).

***Input Validation and Error Handling***  
Robust input validation is a key design element. The function **getValidatedChoice()** ensures that only valid integers within the acceptable range are processed. Additionally, error handling is implemented when reading from or writing to files. For example, if *CS210\_Project\_Three\_Input\_File.txt* or *frequency.dat* cannot be opened, the program outputs an error message and terminates appropriately.

***Screenshots***

* **Figure 1**: Displays the program’s menu interface where the user is prompted for a valid option.
* **Figure 2**: Shows a sample output where the histogram is printed, illustrating how item frequencies are represented by asterisks.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.