Scenario

You are doing a fantastic job at Chada Tech in your new role as a junior developer, and you exceeded expectations in your last assignment for Airgead Banking. Since your team is impressed with your work, they have given you another, more complex assignment. Your task is to build an item-tracking program for the Corner Grocer, which should incorporate all of their requested functionality.

The Corner Grocer needs a program that analyzes the text records they generate throughout the day. These records list items purchased in chronological order from the time the store opens to the time it closes. They are interested in rearranging their produce section and need to know how often items are purchased so they can create the most effective layout for their customers.

Directions

As you work, check your code's syntax to ensure that your code will run. Note that when you compile your code, you will be able to tell if this is successful overall because it will produce an error message for any issues regarding syntax. Some common syntax errors are missing a semicolon, calling a function that does not exist, not closing an open bracket, or using double quotes and not closing them in a string, among others.

The program that the Corner Grocer is asking you to create should address the following functional requirements:

Menu Option One:

1. Prompt a user to input the item, or word, they wish to look for.
2. Return a numeric value for the frequency of the specific word.

Menu Option Two:

1. Print the list with numbers that represent the frequency of all items purchased.
   The screen output should include every item (represented by a word) paired with the number of times that item appears in the input file, CS210_Project_Three_Input_File.txt.
   For example, the file might read as follows:

```
Potatoes 4
Pumpkins 5
Onions 3
```

Menu Option Three:

1. Print the same frequency information for all the items in the form of a histogram.
2. Then print the name, followed by asterisks or another special character to represent the numeric amount.
   The number of asterisks should equal the frequency read from the CS210_Project_Three_Input_File.txt file. For example, if the file includes 4 potatoes, 5 pumpkins, and 3 onions, then your text-based histogram may appear as represented below. However, you can alter the appearance or color of the histogram in any way you choose:

```
Potatoes ****
Pumpkins *****
Onions ***
```

Menu Option Four:

1. Exit the program.

Data File Creation:

Create a data file, with the naming convention *frequency.dat*, for backing up your accumulated data. The frequency.dat file should include every item (represented by a word) paired with the number of times that item appears in the input file.

This output file is created in the beginning of the program without user intervention and is for backup purposes. For assistance with writing files, see sections 7.1, "File Input," and 7.3, "File Output," in zyBooks.

Documentation:

Describe your code's design and functionality. Include screenshots to support your description. As you complete this work, your manager at Chada Tech is interested to see your thought process regarding how you use C++.

Implementation Guidelines

Note: We recommend using the Virtual Lab to complete this assignment. However, if you choose to complete this assignment locally, please use the version of Microsoft Visual Studio outlined in the syllabus. The SNHU IT Service Desk will be unable to support concerns related to local assignments.

- For this assignment, your implementation plan must include at least one **class** with **public** and **private sections**. For assistance with classes, see sections 6.2, "Using a Class," and 6.3, "Defining a Class," in zyBooks.
- For this assignment, we recommend using **Maps** (section 11.4 in zyBooks) for implementation. Maps do require precise coding syntax. However, you can choose a different implementation option.

Apply **industry standard best practices** such as in-line comments and appropriate naming conventions to enhance readability and maintainability. Remember that you must demonstrate industry standard best practices in all your code to ensure clarity, consistency, and efficiency. This includes the following:

1. Inserting in-line comments to denote your changes and briefly describe the functionality of the code
2. Using appropriate variable, parameter, and other naming conventions throughout your code