



CS 210 Project Three Guidelines and Rubric

Competency

In this project, you will demonstrate your mastery of the following competencies:

- Write professional, well-documented, and readable code
- Utilize various programming languages to develop secure, efficient code



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

Optional Challenge: User Input Validation

Validating user input is a coding best practice. For an additional challenge with this assignment, use input validation and error handling to anticipate, detect, and respond to run-time and user errors (for example, make sure you have option four on your menu so users can exit the program).

What to Submit

To complete this project, you must submit the following:

Grocery-Tracking Program

Submit your completed work as a ZIP file, including all Visual Studio project files that are required to run the program. Reference the CS Visual Studio Export Tutorial, linked in the Supporting Materials section, for guidance on how to download the necessary ZIP folder.

Documentation

Submit your completed documentation of your code's design and functionality as a one-page Microsoft Word document with 12-point Times New Roman font, double spacing, and one-inch margins. Include screenshots to support your description. If relevant, any sources should be cited according to APA style.

Supporting Materials

The following resource(s) may help support your work on the project:

Resource: [CS210 Project Three Input File](#)

Use this TXT input file to test your code as you work. It contains a list of different produce items that were purchased on a sample day, beginning in the morning when the Corner Grocer opened and continuing until the evening when the store closed.

Tutorial: [CS Visual Studio Export Tutorial](#)

This guide will walk you through how to download all of your work from Visual Studio as a ZIP folder.

Project Three Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Menu Option One: Individual Frequency	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Create code to process data and output the frequency of a specific grocery item (85%)	Shows progress toward proficiency, but with errors or omissions, e.g., user menu is partially implemented, returning an incorrect or null frequency, extraneous characters (55%)	Does not attempt criterion (0%)	20
Menu Option Two: Overall Frequency	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Create code to process data and output a comprehensive list of grocery item names and their frequencies (85%)	Shows progress toward proficiency, but with errors or omissions, e.g., partially implemented user menu, partial list displayed, missing frequencies, improperly formatted output (55%)	Does not attempt criterion (0%)	20

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Menu Option Three: Histogram	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Creates code to process data and output a text-based, graphical representation of frequencies (85%)	Shows progress toward proficiency, but with errors or omissions, e.g., partially implemented user menu, graphical output mismatch with numerical data, improperly formatted output (55%)	Does not attempt criterion (0%)	20
Menu Option Four: Exit	N/A	Creates code to process user input to exit the program (100%)	N/A	Does not attempt criterion (0%)	10
Data File Creation	N/A	Creates an output file, with the appropriate naming conventions, that contains all accumulated data (100%)	Shows progress toward proficiency, but with errors or omissions, e.g., data file is partially created or file is empty (55%)	Does not attempt criterion (0%)	10
Documentation	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Describes the code's design and functionality with supportive screenshots (85%)	Shows progress toward proficiency, but with errors or omissions, e.g., missing design and implementation details or screenshot instead of written description (55%)	Does not attempt criterion (0%)	10
Industry Standard Best Practices	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Applies industry standard best practices, such as in-line comments and appropriate naming conventions, to enhance readability and maintainability (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include using proper naming conventions or in- line comments (55%)	Does not attempt criterion (0%)	5
Articulation of Response	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose (85%)	Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability (55%)	Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas (0%)	5

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Total:					100%