**CS 499 Module 4 Milestone**

Matthew Ohlemacher

Southern New Hampshire University

# CS 499: Computer Science Capstone

Federico Bermudez

7/27/2025

**CS 499 Module 4 Milestone**

**Algorithms and Data Structures Narrative – Milestone Two**

Artifact Description The artifact I selected for the algorithms and data structures category is a C++ grocery item frequency tracker originally developed in CS 250. The program reads a list of grocery items from a file, counts the frequency of each item, and provides a menu for the user to search, list, and visualize item frequencies. It was created in late 2023 as part of a course project focused on applying basic algorithmic and data structure concepts.

Why This Artifact Was Chosen I chose this artifact for my ePortfolio because it highlights my ability to use standard data structures like std::unordered\_map and algorithms like sorting, counting, and transforming input. Enhancing it gave me a chance to improve not just the performance, but also the robustness and user experience of the application. The original project used std::map, which maintains key order but has slower average lookup performance. I replaced it with std::unordered\_map for better speed, added the ability to sort items by frequency, and improved error handling and file input validation.

These changes helped me demonstrate:

* Knowledge of time complexity and how data structure choices impact performance
* The ability to use sorting algorithms to manipulate and present data
* Input validation to handle real-world edge cases

Outcome Alignment Yes, the enhancements met the course outcomes I intended to target in Module One:

* Apply algorithmic principles: I used hash maps and sorting with lambda expressions to analyze and display frequency data.
* Design efficient solutions using appropriate data structures: I swapped in unordered\_map and structured the code to be more modular and readable.
* Validate and debug data structure operations: I added input checks and tested edge cases to confirm the integrity of the item frequency results.

No updates are needed to my planned outcome coverage.

Reflection on the Process Enhancing this artifact helped me better understand how algorithm efficiency translates into real-world responsiveness. Swapping data structures seemed like a small change, but it required adjusting how I handled the frequency list and histogram. Adding frequency-based sorting forced me to rethink how to structure the data after collection.

The biggest challenge was managing the program flow and making sure user input didn’t crash the application. I had to restructure some logic and test a lot of invalid inputs. This taught me the importance of defensive programming.

Overall, the experience helped me grow more confident in applying data structures and algorithms in ways that make programs faster, more reliable, and easier to use.