4-2 Milestone Three:

Enhancement Two:

Algorithms and Data Structure

Narrative

The artifact that I chose for my capstone encompasses all three categories. It was a program that allowed users to bid on items in a database. It also allowed them to search for and remove items. I created it two years ago in September of 2020 for my Data Structures and Algorithms class at SNHU. The main reason that I chose this artifact is that not only does it encompass all three categories but I felt that there was so much that I could have done better. I really wanted this to be a polished program and unfortunately I ran out of time and chose the easiest route to accomplish my objectives. Now that I have the time I plan to expand upon this artifact exponentially.

I selected this item due to its very nature. It was different data structures accomplishing the same task, and I felt that by combining them into one program and allowing the user to select the data structure they wanted to use would not only allow me to create a “new” program but also allow me to build upon a project that I felt less than great about. By redesigning the three different data structures specifically for this enhancement for Milestone 3 I hoped that it would be a great way to showcase my understanding of Software Engineering and Design, Data Structures and Algorithms, and Databases. This enhancement saw me adding the .cpp and .h files for the two other data structures, HashTable and BinarySearchTree, while ensuring that functionality remained throughout the entire process. An over arching menu was created to allow the user to initially select the data structure type and then from there a secondary menu was created allowing the user to then interact with the data within the Excel files, defined by the data structure.

I feel that the course outcomes were met with what I planned. My initial plan was to consolidate the components of the individual programs into one single program. While this was accomplished, I did find that there was an inherent technical cost to simply combining the individual files without being cautious. Some of the algorithms had to be re-written, such as insertion into the binary search tree, as the algorithm was simply dysfunctional. When analyzing my outcome coverage I am confident that I am within margins for where I intended to be at this point. There are still a few lingering bugs that need to be addressed before the artifact is fully complete.

This process has been interesting. While revisiting old code I found that there were parts that didn’t work properly, or exist at all, and needed more attention than I initially realized. Refactoring the binary search tree took longer than expected and there were some inefficiencies that were introduced as work arounds when dealing with the C++ language. The biggest challenges faced were the amount of build errors and getting all parts to compile and run properly. Debugging and cout statements were a huge help in identifying areas of weakness, or incorrect code.