Brennon Fultz

CS 499

Milestone Three Narrative

The artifact being enhanced for this milestone is the CoursePrinter app from CS 300 Data Structures and Algorithms. This was created in January and February of 2024 and is a simple command line app that loads computer science course data from a CSV file and then stores that information in a binary search tree where it can be printed in order or searched by course ID. This artifact is being included in my portfolio because it targets the differences between two data structures and highlights their respective advantages while using algorithms to perform operations on the data. Proficiency with data structures and algorithms is a skill that employers in the software engineering field are looking to see in applicants, so this project offers an opportunity to showcase those skills. In particular the algorithm to organize the data in the binary search tree and the use of a hash table for faster lookup are the components that best display this proficiency. The addition of the hash table is the enhancement that was made in this artifact, which was done by declaring an unordered\_map in the main function and then updating the loadCourses function to populate this with the same data as the binary search tree. The third case in the switch statement was then updated to call a new searchCourseTable function, which was then implemented using the unordered\_map’s find() method. This allows for courses to be searched by ID in O(1) or constant time complexity, an improvement over the previous lookup speed using the binary search tree.

I planned to meet the third and fourth course outcomes with this enhancement, which require evaluating computing solutions and making design choices, and using skills in computing to deliver value. This enhancement delivers value by improving the lookup speed of the application while showcasing skill in a well-organized and cleanly-coded file. The design choice to add a second data structure to enable improved functionality displays a sound evaluation of computing solutions. With this in mind, I would say that I have met the planned course outcomes for this enhancement. In the process of enhancing this artifact, I learned about the HashTable implementation in C++ which takes the form of unordered\_map, a type I had not used before. In reading the documentation for this data structure, I reinforced my knowledge not only in C++ but also programming languages in general and the similarities and differences in how these languages achieve common abstract data types like HashTables. I wouldn’t say I faced challenges in this enhancement because I had no issues in terms of debugging, everything worked as intended right away. However, this project was one that took plenty of time and effort originally roughly one year ago, so it was satisfying to see how much further I’ve come in terms of coding competency. Below is a screenshot of the program output with both good and bad input.

