Asa Brown

CS-499 Computer Science Capstone | SNHU

July 26th 2025

The artifact is Project Two from the Cs-300 course. The project required me to take information from a csv and sort the courses and their prerequisites into a hash table, then allow the user to look up information after running the program. I wanted to include this artifact in my ePortfolio because I feel that having a solid grasp on algorithms and data structures is a key aspect of programming across multiple career fields. Having this highlighted in my ePortfolio will be very beneficial for showcasing my skills in that regard.

I did meet the course outcome I was aiming for which was #4: Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals. While the system I had in place to view each course and their prerequisites was sufficient to use, I realized there wasn’t an option to print out the sequence of courses that one could take. Through this enhancement, I implemented a means to do this that is also scalable based on the information found within the CSV file. When looking through the artifact, I needed to refamiliarize myself with the source code. Since a lot of the code was already provided by a template, I needed to understand what I built from previously again while trying to remember how I approached a solution to the previous problem, thankfully the code was well-commented so navigating the purpose of each function and variable wasn’t too difficult.

To implement my enhancement, I needed a means to sort the courses in the hash table based on what courses needed to be completed before others and I came across topological sorting and the process involved with implementing it within my code. It required me to make three functions: BuildPrereqGraph(), topologicalSortUtil(), and topologicalSort(). The first iterated through an already made hash table with the course and prereq information in it then puts the information in an unordered map. The Util function then parses through the map recursively and adds the course and the prereqs to a stack, which is then sorted through the sort function. When done it initially printed out the coruses in the reverse order of completion, so I made a vector to store the values, reverse the order then print them off with a newly added case in the main() function. It took a good amount of time to wrap my head around the concept, but online resources like geeksforgeeks and cp-algortihms helped out a ton with my understanding.