1. Briefly describe the artifact. What is it? When was it created?

The original unenhanced artifact is software to help the fictional ABC University’s computer science advisors access course information for students using data structures. This artifact uses a hash table with chaining to deal with key collisions along the features to load data, search for data, and print data from the hash table. This artifact was created on March 5th, 2025.

1. Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved?

I included this artifact in my ePortfolio because it demonstrates my skills in algorithms and data structures through meaningful enhancements. I improved the original project by implementing open addressing with linear probing for collision handling and replacing quicksort with a radix sort designed for alphanumeric course IDs. These changes allowed me to explore performance tradeoffs, handle real world input more accurately, and apply algorithmic thinking beyond standard library functions. The enhancements showcase my ability to design and implement efficient data structures, adapt sorting algorithms for specific use cases, and improve the structure and reliability of existing code.

1. Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

I did meet the course outcomes that I planned to meet with this enhancement in module one. I refactored the hash table to use open addressing and implemented a new sorting algorithm which was radix sort using the course IDs for each course. I have no updates to my outcome coverage plans, everything is still the same as intended.

1. Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?

While enhancing this project, I learned how to implement radix sort to work with course ID strings, which was more challenging than sorting simple numbers. I had to figure out how to handle different string lengths by adding spaces to make all the course IDs the same length, and then sort each character from right to left. After sorting, I also had to remove the extra spaces so the course IDs would display correctly. This taught me how to adapt an algorithm to fit realistic data and how small changes can affect how sorting works. It was tricky at first, but working through it helped me understand string sorting much better.