The artifacts chosen for the Data Structure and Algorithms section of the final project are a Linked List program and a Binary Search Tree program that I created during CS 260. These artifacts were chosen because they came from a course that was based solely on the creation, use, and understanding of data structures and algorithms. These programs show the use of multiple data structures and algorithms specifically tailored to those structures. The programs as artifacts showed by ability to create algorithms to complete tasks for uses and display the results. Each of these artifacts had similar algorithms already including load, search, and delete algorithms that enabled a user to load information from a csv and display it to the screen and then search and delete information. The programs both lacked a sorting algorithm, or a method for which the user to choose to sort their data.

For the Linked List artifact, the enhancement I made was to add an insertion sort algorithm. When the course first started I proposed a merge sort for this program, but realized that as a singly linked list I could do an insertion sort for the program instead. This insertion sort algorithm sorts the information by Bid Id so that the bids are sorted from lowest to highest bid id when the user chooses the sort option. Also during these enhancements, I ensured that I was adding the necessary comments and removing the fix-me comments I noticed during the code review to really clean up the program. This enhancement showcased my ability to not only return to an old program and understand it in depth but showed my ability to program a solution to solve a logic problem in the program. Initially the program just read in the csv and spit out the few pieces of information we needed. By adding this sorting algorithm I could easily build on it to enable sorting by other portions of the bid, not just the bid Id. The enhancement I added to this program did not meet the specific goal I had in mind for creating a merge sort, but it did meet my goal to have a sorting algorithm added to enhance the usability of the program. Although the enhancement was not exactly what I initially planned, it still met the course objective I planned to meet for this part of the final project. I do not have any more updates to make to this program to complete it for the final project, pending any instructor feedback after submission.

The other program I enhanced for this portion of the final project was a Binary Search Tree Program. I initially intended to add a sorting algorithm to this program to increase usability and display my skills, but I realized that the sorting algorithm for this program would be less efficient that updating my current algorithms to allow a user to choose how they would like the Binary Search Tree results displayed. After playing around with sending the binary search tree to an array, and then returning the data to a binary search tree I realized it was going to be more efficient to reload the data depending on how the user wanted to display it. To do this I provided more options for the user on their input page, and updated the insert algorithm to reflect. Now the program default loads the information based on the bid id, but provides the user the opportunity to choose between each of the pieces of information displayed for each bid. To do this I adjusted the insert algorithm originally created for the program to have if/if else statements that determine what factor is used to choose the insertion of each bid. Although the way I went about adding this enhancement was different than originally planned, it still displays my ability to solve logic problems using algorithms. It also demonstrated my ability to look at the efficiency of what I have planned and use innovative skills and techniques to solve the problem. I was able to take what I originally planned and understand a better way to implement for this program. I still met the course objectives I originally intended to meet with this enhancement and artifact.

Throughout the process of enhancing both of these artifacts I continuously took the time to step back and ask myself if what I was doing was efficient and made the most sense. This is a process that I try to implement not just for my programs but my daily life and it paid off through this process because I was able to identify a better and more sound way to achieve my plans for both of these programs. One of the challenges I faced during these enhancements was adjusting my lab environment to be able to work again. Since completing CS260 my laptop has had a couple updates, and one of the updates I made caused Eclipse to not be able to run my programs. There is a compatibility error with Mac and the debugger that Eclipse uses by default, and in the process of keeping my laptop up-to-date I unknowingly caused a compatibility issue that wouldn’t enable my programs to build or debug. I was able to solve this by downloading a more recent version of GDB and create a certificate that enabled the programs to work properly. Throughout the first two milestones I have realized how much more difficult things can be when you aren’t given a lab environment, and it’s made me realize that I need to stay on top of updates and issues that can impact my ability to work on my projects. This was a big lesson for me because most of the classes I was in gave us a working lab environment, so I never had to worry about these things before but now I know what to look for an how to fix several errors.