**4-2 Milestone Three: Artifact Two Narrative**

Kristie O’Brien

Science, Technology, Engineering, and Mathematics Department, Southern New Hampshire University

CS-499 Computer Science Capstone

Dr. Troy Hawk

July 26, 2025

**4-2 Milestone Three: Artifact Two Narrative**

The artifact that I am using for this assignment is Project Two from my CS 300 class. It was created as the final project for that class, and the goal was to work with data structures and algorithms to print course information. It should be noted that when this artifact was originally created, I was still fairly new to computer science as it was completed in one of my earlier classes. Originally, I used a vector data structure and relied nearly exclusively on for loops for this code to print the provided information alphanumerically. This resulted in overly complex runtimes.

I decided to include this artifact in my ePortfolio as I believed it gave me the opportunity to showcase my ability to work with various data structures and algorithms. Having nearly completed my computer science degree, I now know a great deal more about complexity in runtimes, data structures, and algorithms than I did originally. Therefore, I knew I could enhance this artifact quite a bit from where it was in terms of complexity and data structures originally. By changing from vectors to an unordered map for looking up course information, I show how I am specifically able to work with various data structures. By switching data structures, I am also showing how I am able to understand and account for runtime complexities. It should be noted that some aspects of the code, such as looking up course prerequisites, were left with the for loop structure. The reason why is that this allowed for more impactful error handling while not compromising heavily on efficiency. The root of the look up functionality is an unordered map, which keeps the look up functionality efficient.

In addition to changing the data structure to improve efficiency, I also changed the code to make it more readable and maintainable. Originally, my code had two functions, one called UserRun and the other called UserRunTwo, that performed similar functions. However, they had a great deal of overlap and duplicated code between the two of them. As such, I was able to consolidate the functionality down to one UserRun function. I was able to accomplish this when I realized that UserRunTwo was mostly there for various error handling. However, much of the error handling dealt with in UserRunTwo, was handled elsewhere or inefficiently. I could handle such errors in other locations much more efficiently, such as opening the file or within UserRun itself. Therefore, the need for the duplicated code was eliminated.

I believe I successfully met the course outcomes I set out to meet with this artifact. Specifically, the course outcome I sought to meet was course outcome three that states, “Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.”. The reason why I believe I achieved this course outcome is that I was impactfully able to implement algorithmic solutions to computer science problems in my artifact. I specifically implemented algorithmic solutions in my code by working with unordered map data structures to increase the efficiency of the code and decrease the runtime complexity. Therefore, this shows that I was able to impactfully design, evaluate the complexity issue, and create a solution. This solution combined existing code and complexities with new code and complexities to give a cohesive and holistically more efficient code.

I learned a lot more about working with unordered maps in this artifact. I thought I already knew a great deal about unordered maps prior to beginning work on this artifact. However, when I began working with the unordered maps for this artifact, I learned that unordered maps may or may not be the most efficient solution or that they may be combined with other solutions to achieve optimum efficiency. For instance, error handling for prerequisites were still a requirement for this project. As such, prerequisites needed to remain a vector for the most impactful, while still efficient, error handling. However, the course lookup functionality needed to be more efficient such that it was not utilizing a vector. Therefore, I was able to combine unordered map efficiency with error handling to create a much more well rounded and efficient code. Most of the challenges I face while working on this code revolved around finding the best way to balance efficiency with accurate sorting and error handling. I could not compromise on one for the sake of the other. However, after experimenting with a few different methods, I am pleased with the outcome. I believe it balances finding complex information efficiently, such as courses as a whole, while finding less complex information like prerequisites that may be more prone to errors. The process of enhancing this artifact has been extremely rewarding as I believe, compared to last week’s artifact, this artifact was the more complex enhancement. As such, I got the chance to learn more while also improving the code more.