CS 499 Milestone Three Narrative

The artifact that I selected is the Course Manager program originally created during my Data Structures and Algorithms course. Initially developed in C++, I converted it to Python for improved readability and practicality. In my previous artifact enhancement, I used a dictionary to store course data loaded from a CSV file. In this enhancement, I replaced the dictionary with a fully functional Binary Search Tree implementation that showcases my understanding of algorithms and data structures.

I chose this artifact because it provided an opportunity to apply algorithmic principles with an application that simulates real-world data operations. By implementing a BST, I optimized the application for better performance in searching, inserting, deleting, and printing courses. Additionally, I introduced recursive algorithms for the core operations, including in-order traversal and deletion using all three BST cases.

This artifact also served to showcase my software design process and incorporation of secure principles like input validation. Input validation was utilized for course IDs, course duplication detection, and prerequisite rules that check for invalid or illogical course relationships-such as a course being its own prerequisite or having an advanced course as its prerequisite.

My enhancement aimed to meet the course outcomes by introducing a Binary Search Tree and evaluating the trade-off between using a dictionary and a tree structure, and choosing the latter that supports better scalability, improved time-complexity, and sorted traversal. I also used recursive logic, tested various edge cases, and used abstraction and encapsulation to separate various application layers like my CourseManager class, Validation class, and Course class. Additionally, I reaffirmed my security mindset by including validation techniques that prohibit invalid data and enforced logical dependencies. Overall, I believe I sufficiently met the original enhancement goals and aim to continue improving upon the application in the Database milestone.

Through this enhancement, I learned how to design, debug, and implement a data structure essentially from scratch. One of the biggest challenges I faced was implementing the delete() method, which required correctly managing all three required cases of node removal while preserving the tree’s structure. I utilized available online resources like GeekForGeeks and W3Schools to help me grasp and clarify difficult concepts, and I ultimately relied on a step-by-step process for debugging with print statements to trace the behavior and logic.

Throughout this enhancement I learned as I went and now feel more confident in implementing at the least a Binary Search Tree data structure in other applications using Python. While I think I have greatly improved my understanding in data structures and algorithms, I continue to look forward to other data structures and building more applications like this. Additionally, I encountered logical problems like preventing users from adding advanced courses to their prerequisites, detecting and blocking duplicate or circular prerequisites, and also enforcing correct formatting for course ID’s.

Overall, I am greatly pleased with my progress in designing trees and recursive algorithms. I feel it’s forced me to be aware of how data structures are applied in a real-world scenario and the accompanying challenges. In conclusion, I look forward to building future applications in software development, especially those that utilize data structures and algorithms.