Maridelle Gonzales

CS499

**Artifact Description**

The artifact I included in my ePortfolio is a Course Management System implemented in C++, created as part of my CS300 coursework. This system loads course information from a file, stores courses in a hash table for efficient access, and manages prerequisites for each course. It also includes features such as printing all courses in alphanumeric order, displaying detailed course information, visualizing course dependencies through a graph, performing a topological sort to respect prerequisite order, and running unit tests to validate the system’s correctness. The artifact was initially created during the first half of the semester and enhanced over multiple modules to include these advanced features.

**Justification for Inclusion**

I chose to include this artifact because it showcases my proficiency in algorithms and data structures, particularly my ability to:

* Implement hash tables (unordered\_map) for fast course lookups.
* Use vectors to manage lists of prerequisites efficiently.
* Apply sorting algorithms and topological sorting to solve real-world dependency problems.
* Construct a graph data structure to visualize course dependencies, reinforcing my understanding of adjacency lists and graph traversal.
* Write unit tests to verify correctness, demonstrating attention to software reliability and quality assurance.

This artifact clearly demonstrates my problem-solving skills, ability to translate real-world requirements into algorithmic solutions, and capacity to enhance and optimize a system over time. The inclusion of error handling, memory optimization, and graph-based visualization highlights my growth in designing robust, maintainable, and efficient software.

**Course Outcomes and Coverage**

The artifact aligns with the course outcomes I planned to meet in Module One:

* Applying appropriate data structures to manage and process structured data.
* Implementing algorithms to solve computational problems, such as sorting and topological ordering.
* Demonstrating software engineering practices, including testing, debugging, and error handling.

Through enhancements like graph visualization and unit testing, I have strengthened my coverage of these outcomes and improved my understanding of how to translate abstract data structures into practical solutions. I now have a more complete picture of how algorithms, data structures, and software engineering principles integrate in a single project.

**Reflection on the Enhancement Process**

Enhancing the artifact was a highly educational process. As I implemented features like topological sorting and graph visualization, I deepened my understanding of graph algorithms, depth-first search, and course dependency modeling. Writing unit tests taught me the importance of verifying code correctness and anticipating edge cases, while optimizing memory usage and adding error handling emphasized robust software design.

The challenges I faced included managing dependencies between courses when some prerequisites were missing in the file, ensuring const-correctness when accessing data structures, and designing a menu interface that was both functional and user-friendly. By overcoming these challenges, I improved my problem-solving skills and learned to anticipate potential runtime issues, such as invalid input or missing data.

Overall, this artifact demonstrates my ability to create, enhance, and optimize a software system while applying algorithms, data structures, and good software engineering practices which are the skills that are critical for a future career in computer science.