Michael Kibler

CS 499 Capstone

19 June 2025

Narrative

**Artifact: Course Prerequisite Manager**

*Description:*The Course Prerequisite Manager was originally created as a C++ console application during my Data Structures and Algorithms course. It was designed to load a CSV file containing course offerings and their prerequisites, allowing users to view all courses available for enrollment, search for specific courses via their course number, and see any direct prerequisites.

*Justification for Inclusion:*This artifact is included in my ePortfolio because it represents my development as a software engineer, particularly in applying real-world solutions to academic advising through three categories of enhancements:

**Enhancement 1: Software Design and Engineering**Irefactored the C++ version of the course prerequisite manager into a fully object-oriented Java application. This change enabled both better modularity and maintainability. The use of encapsulated classes like Course, CourseGraph, and App demonstrated my ability to implement software design patterns and apply abstraction in my applications. I show strong software engineering principles by adopting class-based organization and separating logic from user interaction.

**Enhancement 2: Algorithms and Data Structures**Instead of the original C++ hash table approach, I implemented a graph-based solution in Java using HashMap with adjacency lists to represent the course structure. I used a recursive depth-first search (DFS) algorithm to determine all prerequisites for a given course. This enhancement demonstrates my ability to apply efficient traversal algorithms to real-world problems.

**Enhancement 3: Databases**To eliminate reliance on volatile memory and manual CSV loading each time a user runs the application, I integrated a persistent SQLite database using JDBC and managed dependencies through Maven. This enhancement demonstrates my ability to implement real-world data storage, providing users with reliable access to course data. It also shows my familiarity with database design and configuration in a production-ready environment.

*Reflection on the Enhancement Process:***Software Design and Engineering**I deepened my understanding of class abstraction, clean code practices, and system scalability while working to convert this artifact into an object-oriented Java application. One challenge I faced was determining how to best separate course logic from UI code, which I solved by refining my class responsibilities. This enhancement helped to reinforce my skills in designing user-focused and maintainable applications.

**Algorithms and Data Structures**Switching from a hash table lookup to a DFS-based traversal involved a deeper exploration of graph theory and recursive algorithms, making me feel much more aware of the capabilities they provide. At first I struggled with ensuring that cycles or invalid inputs didn’t crash the system, but I fixed this by adding validation. This enhancement taught me to more carefully manage all edge cases and recursion depth, and helped with my grasp on algorithm design.

**Databases**Integrating SQLite required researching and testing Java JDBC connections, crafting a schema that matched the previous CSV logic, and validating queries to provide better security. The biggest challenge for this enhancement was understanding how to sanitize inputs and manage exceptions. I learned how to configure Maven to manage the SQLite dependency needed for JDBC and improved my understanding of persistent storage.

*Course Outcomes Met:*

* Demonstrated collaboration and communication during peer code review planning and walkthrough videos.
* Delivered professional-quality software with structured classes and logical flow.
* Applied algorithmic principles using depth-first search in course sorting.
* Employed reusable design patterns (encapsulation, separation of concerns).
* Validated inputs and implemented exception handling, demonstrating a security-conscious mindset.
* Integrated persistent database storage with JDBC and Maven, aligning with industry software development practices.