CS-499: Computer Science Capstone

3-2 Milestone Two: Enhancement One: Software Design & Engineering

Emily Black

November 14, 2025

ARTIFACT DESCRIPTION

The artifact used in Enhancement One is my Course Planner program from the CS-300 Data Structures and Algorithms course, which is a C++ console application that reads course and prerequisite data from a CSV file and supports printing a sorted course list or viewing a single course by ID. The original version was structured procedurally in one file and used a global container for the data, several helper functions, and a large main function to coordinate all behavior.

ARTIFACT INCLUSION JUSTIFICATION & IMPROVEMENTS DESCRIPTION

I included this artifact because it reflects a common data task that a data analyst encounters, which is ingesting structured data, organizing it for efficient retrieval, and presenting it to a user. The original code worked, but it mixed user interface logic, business logic, and data access in one place. In this enhancement, I refactored the design into four classes: 1) `Course` encapsulates domain data, 2) `CourseRepository` manages loading from CSV and provides lookups and sorted views, 3) `CourseService` prints lists and details and will host future domain logic, and 4) `MenuUI` handles console interaction and input validation. I also added basic exception handling for critical failures and warnings for malformed lines, and I documented the code with inline and Doxygen-style comments to clarify responsibilities and intent. These changes align with code review best practices that emphasize clarity, structure, and maintainability (SmartBear, 2024; Pluralsight, 2024; SNHU, 2024), and they follow secure coding guidance about validating and constraining input before use (OWASP, 2024).

COURSE OUTCOME COVERAGE

This enhancement addresses the Software Design and Engineering course outcome, and now, the code demonstrates separation of concerns, encapsulation, and clean interfaces across layers. It also supports algorithmic thinking by selecting structures that fit the access pattern. For example, the repository uses an unordered map to support constant-time average lookups by ID. The approach is consistent with research that connects structured review and incremental improvement to better long-term software quality and clearer ownership boundaries (Fang et al., 2023). I plan to address database integration in a later milestone where I will introduce parameterized SQL queries and indexing to improve security and performance.

PROCESS & CHALLENGES REFLECTION

Refactoring required me to trace the original data flow and decide where to draw boundaries between data access, business logic, and presentation. The most significant challenge was adding error handling without complicating the control flow. I resolved this by throwing exceptions for critical failures, such as file open errors, and logging warnings for non-critical anomalies, such as malformed lines that can be skipped. I also found that documentation and naming were important for maintainability. The combination of comments and smaller, focused classes made it easier to validate the design and to explain it during my code review. This enhancement reinforced how professional code reviews are not only about finding defects but also about improving communication and maintainability for the long term (SNHU, 2024; SmartBear, 2024).

REFERENCES

Fang, Y., Wang, J., Wu, Y., Liu, H., & Liu, S. (2023). CodeCV: Mining expertise of GitHub users from coding activities. *IEEE Access, 11*, 1262–1275. https://doi.org/10.1109/ACCESS.2023.3232918

OWASP. (2024). *Code review guide*. OWASP Foundation. https://owasp.org/www-project-code-review-guide/

Pluralsight. (2024). *Code review checklist: Best practices for reviewing code*. https://www.pluralsight.com/blog/software-development/code-review-checklist

SmartBear. (2024). *Best practices for peer code review*. https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/

Southern New Hampshire University. (2024). *CS-499 code review checklist* [Transcript]. SNHU Media. https://snhu-media.snhu.edu/files/course_repository/undergraduate/cs/cs499/cs499_code_review_checklist.pdf