**Artifact Enhancement Narratives for Capstone Project**

**Artifact 1: ABCU Course Planner (C++)**

**Category: Software Design and Engineering & Algorithms and Data Structures**

This artifact is a console-based course planning system built in C++ using a binary search tree to organize and manage course information. I created it during my CS 300 course as a way to apply tree-based data structures to solve a real-world advising problem. The system allows users to load data from a file, display courses alphabetically, and view detailed course information with prerequisites.

I selected this artifact because it highlights both my understanding of algorithms and my growth in clean, modular software design. The enhancements improved the file parsing logic using stringstream, corrected prerequisite validation by implementing proper tree traversal, and removed unnecessary use of dynamic memory allocation. I also added clearer user prompts and input validation, showcasing thoughtful design for usability and safety. These improvements demonstrate my ability to implement data structures efficiently and create user-friendly solutions.

While enhancing this artifact, I gained deeper insight into binary search tree operations and memory management. One challenge I faced was identifying and correcting logical flaws in how prerequisites were validated. I incorporated feedback by breaking complex tasks into smaller helper functions and improving code readability. This artifact aligns strongly with course outcomes in algorithm design, structured problem solving, and secure software development. I feel that I met all relevant outcomes with this enhancement.

**Artifact 2: GameService Singleton (Java)**

**Category: Software Design and Engineering**

This Java artifact implements a singleton class that manages game instances for a simulated game application. It maintains unique game objects, along with player and team IDs. I originally created it in my object-oriented programming course to demonstrate design patterns, encapsulation, and controlled object creation.

I included this artifact in my ePortfolio because it reflects my grasp of core software engineering concepts like object management and scalability. For the enhancement, I refactored the addGame() method to reuse existing logic from getGame(name) and replaced index-based loops with Java’s enhanced for-loops. I also encapsulated static fields more appropriately and followed Java naming and commenting conventions for maintainability. These updates improved the design consistency, made the code easier to read, and aligned it more closely with industry standards.

Through this enhancement, I learned the importance of writing clean, DRY (Don’t Repeat Yourself) code. A key challenge was maintaining the integrity of singleton behavior while improving structure. I applied feedback related to modularization and maintainability. This artifact supports course outcomes tied to design quality, professional coding standards, and communication through clear documentation. I believe all targeted outcomes for this artifact were successfully met.

**Artifact 3: Animal Shelter MongoDB Script (Python)**

**Category: Databases**

This artifact is a Python script that connects to a MongoDB database and retrieves animal shelter data using the pymongo and pandas libraries. I originally created it to demonstrate simple data extraction from a non-relational database for analysis. It loads data from the AnimalShelter collection and prints it in DataFrame format.

I chose this artifact because it shows my ability to work with databases and cleanly handle external data. I enhanced the script by modularizing the database connection logic into a reusable function, replacing print statements with Python’s logging module, and encoding credentials securely using quote\_plus(). I also added error handling to prevent crashes and implemented safe fallback behavior when data fails to load. These improvements helped make the script more secure, professional, and user-friendly.

Enhancing this script taught me valuable lessons in Python scripting best practices, modular design, and secure handling of sensitive credentials. One challenge was refactoring the code to remain clean and readable while introducing logging and error handling. I applied best practices from feedback I received in earlier scripting projects. This artifact meets the course outcomes related to database access, data integrity, and secure code development. I feel confident that the enhancements demonstrated all intended outcomes.

**Conclusion**

Each of these artifacts represents a significant step in my growth as a developer. The enhancements I made demonstrate my ability to refine, maintain, and secure code effectively. Together, they highlight my readiness to work in professional software development environments and show mastery in software design, algorithms, and database technologies. These narratives provide insight into my problem-solving process and reflect how feedback, research, and hands-on experience contributed to my development.