**📄 CS 499 Milestone Two Narrative**

**Name:** Caeli Firth  
**Category:** Software Design and Engineering  
**Artifact:** 3D Study Space Rendering Project (Original CS 330 Project)

**Artifact Overview and Justification**

The artifact I selected for the Software Design and Engineering category is my 3D Study Space Rendering Project, originally created in CS 330: Computer Graphics and Visualization. The project is a procedural OpenGL scene that visualizes a small desk, chair, lamp, and surrounding furniture in a virtual room. While the original version successfully rendered a visually appealing scene, the architecture lacked scalability, reusability, and secure design principles. As part of my enhancement plan, I restructured the codebase to follow object-oriented programming (OOP) principles, introduced modular design, improved documentation, and implemented basic error checking for OpenGL functions.

I included this artifact in my ePortfolio because it allowed me to demonstrate fundamental software engineering practices that reflect industry expectations. The enhanced version shows how I can take a working prototype and refactor it into a maintainable and professional-grade solution. Specifically, the artifact now contains encapsulated classes for each scene object (e.g., Desk, Chair, Lamp), separated input and rendering logic, and documented functions that explain their purpose and implementation. These enhancements display my ability to apply modern design techniques, follow best practices, and make informed engineering decisions—all of which are critical in a real-world development environment.

**Course Outcome Alignment and Updates**

My enhancement work demonstrates progress toward several CS 499 outcomes. I addressed **Outcome 2** by creating professional-quality written and visual documentation, including comments and architecture diagrams that explain system structure. I met **Outcome 4** through the use of OOP refactoring, modular breakdown of logic, and the introduction of reusable class structures. Additionally, I made progress on **Outcome 5** by implementing OpenGL error checking and organizing the application flow to avoid runtime vulnerabilities or silent failures. These steps reflect a security-minded approach to software development.

There are no major changes to the outcome-coverage plan I proposed in Module One. However, I’ve found more opportunities to refine **Outcome 1** as I further formalize architectural diagrams and integrate computing theory into future enhancements.

**Reflection and Learning Experience**

Enhancing this artifact was a valuable learning experience in both theory and practice. Refactoring a procedural program into an object-oriented architecture challenged me to think critically about separation of concerns, inheritance, and reusability. I learned how important modular design is for debugging and testing, especially when projects scale beyond a single file or function. Adding OpenGL error checking taught me how to think defensively when programming graphics applications, where failure points are often subtle and hard to trace.

One of the biggest challenges I encountered was identifying redundant code patterns and designing class hierarchies that avoided code duplication while supporting extension. I also had to manage OpenGL resources carefully to prevent memory leaks during object instantiation and cleanup. These challenges helped me grow in my understanding of both low-level graphics programming and high-level design strategy.

This milestone has not only improved the technical quality of my artifact, but also helped me practice articulating design decisions, documenting solutions, and preparing code for presentation to future employers. It was an essential step toward producing a portfolio that showcases the depth of my skills as a software engineer.