**CS 499 Module 3 Journal**

Matthew Ohlemacher

Southern New Hampshire University

# CS 499: Computer Science Capstone

Federico Bermudez

7/20/2025

**CS 499 Module 3 Journal**

**Software Design and Engineering Narrative – Milestone Two**

**What the Artifact Is**I picked a C++ file encryption project I originally built for CS 405: Secure Coding. It was done earlier this year and shows how to apply basic secure coding with XOR encryption. The program reads a file, encrypts the text using a key, and saves the encrypted version. It also decrypts that data and writes it back to a file.

**Why I Chose This Project**I added this to my ePortfolio because it covers key software design concepts like breaking up code into pieces, reading/writing files, and using a simple encryption method. Updating the project gave me a good chance to clean up the design and apply better practices like separating logic into functions, checking if files open properly, and logging important steps.

When I first wrote it, everything was mostly inside main(), and there were no checks for file errors. There was no feedback if encryption or decryption failed. These were things I fixed with the enhancements.

Here’s what shows my skills:

* Moved the XOR logic into its own function to reuse easily
* Added file checks to catch problems early
* Logged when steps finished or failed using a simple text log

These updates made the code easier to follow and more like what you’d see in a real project outside of school.

**Outcomes I Met**I stuck to the outcomes I picked in Module One:

* Used secure coding: added file validation and better structure
* Showed solid software design: made reusable functions and improved layout
* Applied engineering techniques: added logging to track actions

No changes to my outcome plan I think this artifact covers it well.

**What I Learned**Working on this helped me realize that writing code isn’t just about making it run it’s about making it clean, easy to update, and safe. Moving the encryption part out of main() made me think more about reuse and testing. Adding logs helped me see how to trace what the program is doing, which is huge for fixing bugs or handling real users.

The hardest part was changing how the code worked without breaking it. I had to test a lot to make sure it still encrypted and decrypted the same way after the changes.

In the end, this project reminded me why clean, simple, and testable code matters because it’s easy to use.