**1. Briefly describe the artifact. What is it? When was it created?**

This artifact is a simple console app for a client management system for an investment company. I originally derived it from part of a reverse engineering assignment where I had to look at assembly code and figure out what the original C++ program did. The program keeps track of five clients and whether they chose brokerage or retirement services. It also has a simple password system to control access. This project was created in CS 405 last quarter

**2. Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

I picked this artifact because it really shows how much I have learned about writing good code. The original version was pretty messy with global variables everywhere like name1, name2, var1, var2 and etc. I had about 15 global variables just sitting at the top of the file. The new version uses the Singleton pattern which is a neat software engineering design pattern that ensures only one instance of something exists. I put all the client data into a nice structure and array instead of having separate variables for everything. I also moved all the code into a class so everything is organized in one place. The old version had the same if statement repeated five times for each client, but now I use a simple array that's much cleaner.

**3. Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

The enhancement shows I can use design patterns, write object oriented code, and turn messy code into something professional. I dont need to change my plans since everything worked out as intended.

**4. Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

Working on this taught me a lot about the Singleton pattern and why it's useful. I didn't know before that in C++ you have to define static variables outside the class, which was confusing at first. The hardest part was figuring out what the original code was trying to do since the variable names from the reverse engineering didn't make much sense. I had variables called var1 through var5 and had to figure out they were actually storing service choices.

I also learned how much better code can be when you use proper design. The original had over 50 lines of messy globals and separate functions, but the new version is much easier to understand. It was satisfying to see how organizing everything into a class made the whole program clearer. The main challenge was keeping it simple while still doing it right, since I wanted to show I understood the pattern without making it overly complicated.