**3-2: Milestone Two**

Dylan Jeffery

SNHU

CS-410

Brooke Goggin

3/18/2022

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

This initial artifact (originally “Investments.cpp”) is from CS-410 Reverse Software Engineering. The original file is a C++ file that was created by disassembling a binary file and eliciting program requirements based on the translated assembly code. The file was then reconstructed as the provided “Investments.cpp” file. For my enhancement I have decided to convert this program into Java, and to further enhance this program I will implement a module that uses getters/setters to create a more dynamic application. The current iteration still relies on static global variables, but I would like to develop it further to improve its functionality and security.

1. 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 chose to include this artifact for a few reasons. The first being that it exhibits a technical skill set (eliciting program requirements solely from binary executables is not a simple task) as well as the fact that converting from C++ to Java, further validates my flexibility in a development environment. These are my two most preferred languages, but this artifact is my first time really transferring one into the other. Upon initial translation there were a few minor snags, but I have since resolved these issues using Java best practices and believe that this artifact showcases a deep understanding of being able to break down, re-purpose and re-invent software projects.

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

My plans for enhancement were initially very limited. In Module One my proposed enhancements for this artifact were solely to work out fringe security cases, and to incorporate a more robust login system. I reached out to my instructor and as a modified proposal, decided to convert the program to Java and work from there on further enhancement. As Java is a language that affords us a lot of freedom when it comes to modular design, I intend to continue working on and polishing this artifact and create accompanying modules to better handle user choice as well as a better login system.

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

As this was my successful venture into converting C++ into Java, I would consider it a learning experience. The biggest lesson for sure, had to be regarding the difference in designing security between C++ and Java. In C++, overflow errors had to be designed for explicitly. Each case that could result in overflow had to be considered and user input that failed, meant that behind the scenes management had to be done, and buffers had to be cleared. In Java most of the security was handled efficiently with try{} catch{} statements. This translation wasn’t an obvious conclusion, but rather something I had to understand through trial and error. In further enhancing this artifact I am sure that I may run into vulnerabilities that will have to be explicitly dealt with.