**CS 499 Milestone Two: Enhancement One: Software Design and Engineering**

**By Alexander Varljen**

**July 17, 2022**

**Description of the Artifact:**

The artifact demonstrating my software development/engineering skills, which was created between July 10 and July 17, 2022, is a Python reconstruction of a banking application’s C++ source code, which is from a project that I had worked on in a previous course. This artifact presents a menu which asks the user to give an initial investment amount, a monthly deposit amount, an annual interest rate, and the number of years (up until five years) in which they wish to see the change in their end of year balance and accrued interest. Then, the calculations are made based on the users input and an output display is made.

**Why This Artifact was Chosen:**

I chose this artifact because it demonstrates my Python programming skills, which is a language commonly used in data analysis and data science, my main fields of interest. The range of Python programming techniques employed in this enhancement include user defined functions for subprocesses such as checking and receiving valid inputs, input receivers, while-loops, conditional statements, output displays, and basic arithmetic operations for calculating the year-end balance and interest. Also, through working on this artifact, I was able to secure the integrity of the software more faithfully by including exception handling for user inputs, which was not included in the original C++ project. This demonstrates my ability to develop more secure, functional, and reliable software.

**Artifact Enhancement and Course Objectives:**

My main objectives for this specific enhancement where to take my old project, give like functionality to the source code written in Python, make it more secure, and to add clearer and more concise code comments for the benefit of other developers and reviewers of the source code. I believe that I met all those objectives while working on this artifact, as this program has the same functionality as the old C++ project and multiple try-catches were employed for exception handling around areas that accept input and do calculations. There were also four course outcomes for this enhancement that I was to try and achieve, which are the following:

1. Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science.
2. Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.
3. Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals (software engineering/design/database).
4. Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

With regards to achieving the first of these four course outcomes, I was able to create many descriptive code comments throughout the program explaining the programs functionality, create a description of the program at the top of the source file, and add modularity by using various user defined functions throughout the program. These features grant programmers and reviewers the ability to look through the program and make sense of it, which can make it easier for them to change, add to, or understand the program. In terms of reaching the second course outcome, this document, as well as an in-program code comment at the top of the source file providing a description, seeks to explain the accompanying Python banking program artifact using accurate and comprehensive language. In terms of reaching the third course outcome, I was able to create a working program that does what it is meant to do, which is to take in four user inputs, use them to calculate end of year balance and interest, and display the results to the user. Also, I sought to use well-founded techniques, such as creating user-defined functions to handle subprocesses and exception handling to check for potential errors and bugs. These cases of exception handling, which can be found in all functions that take in input and perform calculations, also aided me in achieving the final of the four course outcomes.

**Reflection:**

In the process of enhancing and modifying the artifact, I’ve had to refresh my knowledge of Python, as most of my previous experience with Python has been for constructing middle layer code for database queries and for statistical analysis. Therefore, I had to take some time and read documentation and practice with Python for more general purposes. Also, much of the new information that I needed to learn was on exception handling in Python. I’ve had previous experience with exception handling using Java, but never with Python before working on this artifact. These challenges have caused me to, initially, work slowly on this artifact as I got my bearings. However, I’ve got more confident and competent along the way as I continued to work throughout the week of this artifact’s completion.