**CS 499 Artifact One Enhancement: Bank Program**

**By Alexander Varljen**

**August 11, 2022**

**Description of the Artifact:**

The original project that I picked to demonstrate my software development/engineering skills, which was created in February 2022, was the source code of a banking application called “CS 210 Project Two”. This artifact presented 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 to five years) in which they wish to see the change in their end of year balance and accrued interest. Then, the end of year balance and accrued interest are calculated based on these inputs and an output display is made. My enhancement of this artifact was to recreate the program using Python, as well as to add more security measures and shorten the length of the program’s source code to increase efficiency. This enhancement was made during the second week of July in 2022.

**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 areas of the code receiving user input, as well as in areas that perform calculations.

**Course Objectives:**

While working on this artifact enhancement, I was able to employ strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science. This was done in three main ways. First, there are links to the zipped source file of this enhanced artifact, as well as to the original C++ project, which are both on my public GitHub repository. That public page also has a readme explaining the purpose of those and other files, along with an invitation to download any of the files for oneself. This allows anyone wishing to participate to learn about or potentially add to this project and make it their own or contribute to the project for the benefit of any developers accessing the page. Second, the source code in the enhancement itself employs modularity and clear user defined functions. This allows others to easily add to, remove, or move these modules, as well as more easily navigate the source code, for the purposes of collaborative work or independent efforts. Third, I made descriptive and specific code comments that allow others who would want to understand, change, or add to the code more easily able to do so.

While working on this artifact enhancement, I was able to design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts. In terms of oral communication for this artifact, I covered the original artifact in the first part of the code review video, where I went module by module discussing what the functionality of each module does and what improvements could be made. In terms of written communication, I wrote thorough code comments in my enhancement’s source code that describe the functionality of the user-defined functions and variables throughout the source code. I’ve also written this accompanying narrative to describe why I’ve chosen the artifact, what my enhancement of it was, and why I chose this enhancement. In terms of visual communication, I created an accurate flowchart showing the basic functionality of the artifact.

While working on this artifact enhancement, I was able to 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. One of the well-founded techniques that I employed in the creation of the source code for the artifact enhancement was to use exception handling in areas of the code that receive user input and perform calculations. This helped ensure that specific messages were presented by the program if a bad input or an error in calculation has occurred. Also, by creating a program composed of modular user-defined functions, I was able to demonstrate the well-founded technique of making a program modular. The tool that I used to develop, test, and run my artifact enhancement’s source code was the Python 3.6.6 Shell. This tool allowed me to run lines of code one module at a time to watch for errors and output in specific areas of the code, which helped me ensure that, by the end of my work, there were no errors and that the output is what was intended.

While working on this artifact enhancement, I was able to 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. There are no semantic or syntax errors in the enhanced artifact’s source code. I was able to ensure this by running the program module by module using the Python 3.6.6 Shell. Also, I was able to add exception handling and messages for multiple situations involving invalid inputs and calculation errors. For example, for the user-defined function containing the code for receiving the initial investment amount from the user, I created exception handling code that throws out the message “Invalid initial investment” if the values are less than or equal to zero, as only a positive numeric value would make sense for the program’s purposes. I created similar exception handling code for the other three inputs asking for numeric values, as well as for the areas performing calculations based on the inputs.

**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. Much of the new information that I needed to learn was also 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.