Kelly Perez

11/16/2024

CS 499

Milestone Two Narrative

**Intro to the Artifact**

I'm showcasing an artifact that was once created in C++: a bank simulator program. It calculates interest with and without further monthly deposits over a predetermined number of years, stimulating the growth of an original investment. This artifact was part of a software development course to practice financial computations and object-oriented programming.

**Justification for Including the Artifact**

I selected this artifact for my ePortfolio because it effectively demonstrates my foundational skills in software development, particularly in using object-oriented design to build functional, real-world applications. The artifact showcases my ability to organize code within a class structure, handle user input, and perform calculations clearly and efficiently. Additionally, it highlights my ability to manage and format output in a readable format using precision control. The artifact has been improved through better naming conventions, enhanced modularity, and refactoring to reduce code duplication, making it more maintainable and scalable.

Course Outcome Alignment

Many of the course objectives, which focused on developing scalable and effective computing solutions, I was able to accomplish. I improved the code's readability and performance, especially by revising the methods and adding error handling to fortify the program. The new artifact now clearly demonstrates my abilities to write software with cleaner, more modular code and by best practices. I don't currently have any changes to my plans, and I believe I've achieved my initial goals in terms of outcome coverage.

**Reflection on the Enhancement Process**

Enhancing and changing the item was a worthwhile educational process. I learned more about how minor design adjustments, such as improved variable name and code modularization, can have a big influence on the readability and maintainability of code. In order to make the program more user-friendly, I also learned how crucial it is to incorporate error handling and to account for edge circumstances like invalid user input.

**Challenges Faced**

Making sure the financial computations were precise and effective when converted to Python was one of the difficulties I faced, especially when handling precision for monetary quantities. Careful testing and validation were also necessary to restructure the code without adding new bugs. I became more aware of the significance of developing clear, testable code after debugging and confirming the results of the program in both Python and C++. In conclusion, this artifact is a powerful illustration of my evolution as a software developer since it not only demonstrates my technical proficiency but also my capacity for self-reflection and work improvement.