**My artifact**

For my software design and engineering artifact, I chose to use a Java based console application that adds and tracks service animals, specifically dogs and monkeys. I created this originally in IT 145, and it enabled users to add in, store, and find animal information like their breed/species, training status, and service country. The original version had some input validation that was limited in scope, was inconsistent in its design, and had little to no error handling.

**Justification**

I decided to use this project because it had object-oriented principles, but also offered a lot of room for improvement. My enhancement plan gives me the opportunity to display some essential software development skills. I chose to make the design more organized and efficient by refactoring my logic into different classes to keep the Driver.java clean and also added in several helper methods to handle the shared validation between the dog and monkey classes.

I wanted to add in more validation which was very limited previously so that the user input is accurate. The code now checks that the date is in the correct format, that the integer or decimal values are the right type for the value and that the species for the monkeys and the training statuses match the criteria I’ve laid out, which ensures the system’s integrity. Essentially, I added in validation for every question to make sure the system is as accurate as possible. I also added in error handling with the validation so that the user understands exactly what is wrong with the input to improve the intake process and be more user-friendly.

Finally, I made sure that my code and commenting is consistent and uses the same naming conventions across the various classes to provide easy to read code. Overall, I think this enhancement showcases code maintainability, design principles, and usability which are all key components of software development.

**Course Outcomes**

* Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.
* 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.

I believe I met the course outcomes that I set out to meet by delivering reusable elements and foundational design principles to keep my code concise and professional. I utilized input validation and flow structures to demonstrate my ability to solve real-world problems using standard programming. I made sure to keep my code well-documented throughout so that it’s easy to understand which makes it maintainable for any other software developers that would work on this project. I think I’ve made strong progress toward both my outcomes for this enhancement.

**Reflection**

I learned a lot about my code and from my code throughout the enhancement process. I hadn’t worked with this project since IT 145 so although I did a code review, it took me a while to get back into the project and understand the best way to apply my enhancement. I had to try to account for a variety of outcomes, and I came up with more and more validation and error handling as I went through the code as I found a bug or something that didn’t make sense from a user perspective. For instance, my original code did not allow for an exit from the animal intake which is a flaw for user experience so I had to make sure I had a helper method to handle that for each step of the intake process.



Another challenge that I faced was checking the different types of input that I had throughout my code including integer, boolean, decimal and others. I wanted to make sure they could all use the same quit logic, and try to combine what validation I could since several of them were the same. I reconsidered and refactored my code many times to get it to be the most efficient. This will help to simplify the code and reduce the likelihood of bugs in the program. This taught me how to rethink code and the various responsibilities, which makes it more maintainable.

