**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 code did not include input validation, error handling, and additional sorting for adopters.

**Justification**

I chose the Rescue Animal artifact because it had good foundational code and followed object-oriented principles. By adding in error handling and input, I improved my code in enhancement one but I wanted to make some additional changes that would display my ability to improve my data structures and algorithms. By adding in some conditional logic and lists, I was able to add in some matching logic.

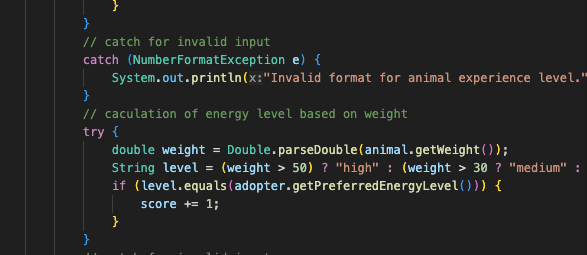
To display my skills in my artifact, I added in a few elements. I created a MatchScorer class that uses weighted scoring so I can now match the adopter with a rescue animal on a ranked scale. The ranked scoring uses the ScoredAnimal and my list to sort the matches in descending order. To continue to improve my structures overall I added in validation for the user so they can’t input a species, experience level, and energy level that is not allowed. The animals are also sorted so that only dog or monkey results would display in the ranking depending on what type of animal the adopter is looking to find. My enhancements make my code more functional and offer more features for the user.

**Course Outcomes**

* Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.
* 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 my planned course outcomes by demonstrating my ability to improve data structures and algorithms. For outcome one, I designed my compatibility match and used object-oriented principles. I also refined my solution by adding in specific filters and validation for each. I tried to keep everything focused and simple so that it would be easy to manage. For outcome two, I met the objective by using the array list to store my dynamic list and used my sorting algorithms to display my technique.

**Reflection**

In enhancement one, I had to redo my code and separate everything out by adding validation but it was essentially the same code that I just improved. For this enhancement it was very different, since I had to add in new code functionality, I had different challenges. I learned a lot about exception handling through this process to make sure the user input is correct. I also learned how to implement scoring logic, which is not something I had done before. It isn’t perfect; I had to base the energy level on weight which is not always indicative, but I couldn’t think of another way to determine the energy level.

One challenge that I faced was making sure that my top 5 results only displayed the species that matched what the adopter had added as input. I had to filter and validate based on what they entered and ensure that the algorithm was sorted correctly. I also had to check that my input validation wouldn’t crash the program, which I ended up using continue statements for. I also added in error handling that allows the user to exit at any point. Overall this enhancement taught me a lot about the different ways that algorithms can be used to improve the user experience.

