Enhancement One: Software Design/Engineering

The artifact that was chosen for software design and engineering is project 1 from foundations in app development class. Project one is an application to check in pets for boarding. The two types of pets are cats and dogs and the pet’s information, number of days they are staying and the cost of boarding. This application was created using java. There are multiple classes that were created for this project. There was a driver, Dog. petClass, and Corgi. Before enhancements this application did not have the ability to check in cats for boarding which is a requirement for this application.

I chose this artifact for the ePortfolio because it showed an application with the java language. It shows an application that requires multiple classes. It also shows an example of parent and child classes. It also shows how child classes inherit variables and methods from the parent class. This application also had enhancement that would noticeably improve the quality of the application. The first enhancements were correcting previously missed mistakes in the program. The Dog class was meant to be a child class of the petClass but when initially created it was not. This was correct by making it extend the petClass. Comments that were missing in petClass so they were added. While not an error it is a best practice to comment code for clarity. The Dog constructor was edited to match the Corgi one. The Corgi class is a subclass or child of the Dog class. This ensured both constructors worked. Next enhancements added capabilities that were not previously in the application. I then created the Cat class to allow cats to be checked in. The cat class is similar to the dog class but does not include the topTrick variable due to cats not typically performing tricks. There are breed, type, and name attributes due to cats having these traits like dogs. Cat was also created to be the child class of the petClass. The Siamese class was created as a child class of cat class. Similar to the Dog class having a Corgi as a child class to it. The Siamese class was created similar to the Corgi class. The Siamese class has the traits of the parent class Cat. Similar to the Corgi class there are the weight and age variables. Another enhancement was adding getters to the setters that did not have them previously in all of the previously created classes and the new classes.

This narrative meets the course outcome “Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts”. This is because this narrative is designed to allow employers to understand the enhancements made to the project. This narrative is technically sound and written in an informal and professional manner. The enhancements I made to project one meets the course outcome “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”. The enhancements I made to project one demonstrates this outcome by ensuring the requirement for both cats and dogs to be able to be checked in for boarding can be accomplished. The use of making parent and child classes to ensure pets can be boarded while specifying if the pet is a dog or cat and what breed they are. Value is delivered because the application meets the requirements specified by the client.

The enhancement process was an interesting learning experience. I learned that most applications can be improved and fixed. Sometimes looking at a project after not working on it for a little bit makes it clearer what errors were made and improvements that can be made. One challenge faced was changing the petClass method setPetType() to only accept pet types of cats and dogs. After a few tries I decided and if branch was the best way to accomplish this goal. A loop was considered first but an if branch ensures only the proper input will allow this method to successfully execute. If invalid input is provided by the user, then the method fails. Another challenge was ensuring changes made did not negatively impact other classes, other parts of code and ensured the application