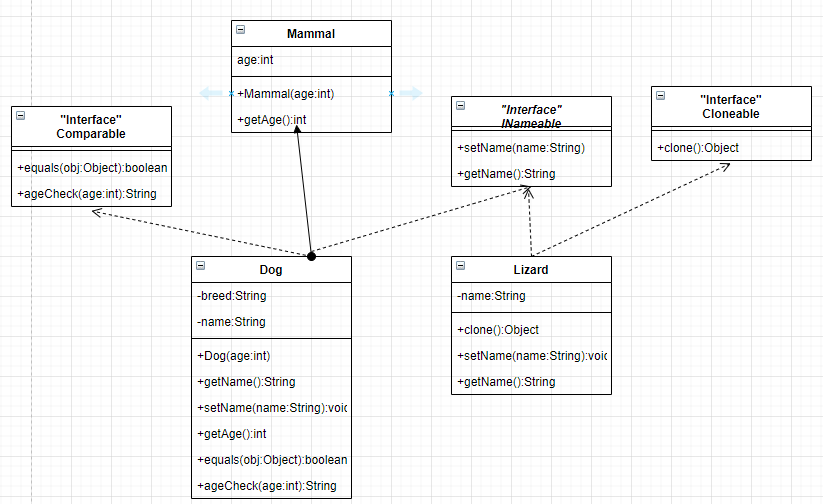
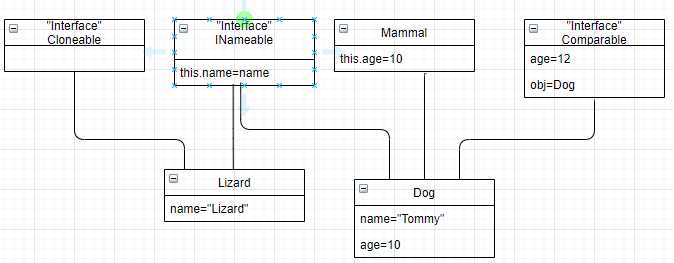
Question 2: Reflection of Assignment 2

Q2.1:

UML Class Diagram



UML Object Diagram:



Q2.2:

In this assignment, we used the concept of polymorphism; overriding methods.

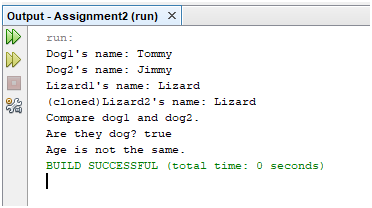
INameable Interface is implemented by Lizard and Dog. Both Lizard and Dog need names so they ”inherit” from that interface. Since interface cannot contain arguments, we need to override those abstract methods.

A dog is a mammal, so it inherits Mammal class. It shares the same value; age.

The Cloneable interface is implemented by Lizard. This interface helps to clone an object, therefore it requires “an exception” in the abstract method. When overriding the method, we keep the return type as “Object” and do casting on the main method. The reason being is to make this method widely usable by other objects instead of getting stuck to one type of Object.

The Comparable interface is implemented by Dog. We added two functions; compare object type and age. If we want to check whether a cat is a dog, we can do a comparison by calling the “equals()” method. The method will check whether this object is an “instanceof” Dog object. In the end, it returns an answer “false”; a cat is not a type of Dog. Age comparison, it must be the same type of object(Dog) and compare the age of each other.

Q2.3 Project run



Q2.4.

In the beginning of the coding, we faced some errors. The concept of interface between the classes is somewhat complicated. We searched for examples of the interface, their meaning, and how to implement correctly.

We used inline commenting because it helps my partner and myself to understand the purposes of the classes, methods and the code easily.

For testing strategy, we prefer to do line by line testing, which helps us to know whether the code is executed or not. When we had no more idea about the errors, we tried to find the similar code and examined why it works and we learnt from it.

For code optimization, we tried our best to reduce the lines as much as possible. For example, when we return an object, we do casting the object within a line. And also, we try to reuse the local variables instead of creating unnecessary ones.

From this assignment, we learned that what is the purpose of interfaces and the effect of interfaces on the different classes. It makes the program easier and understandable.