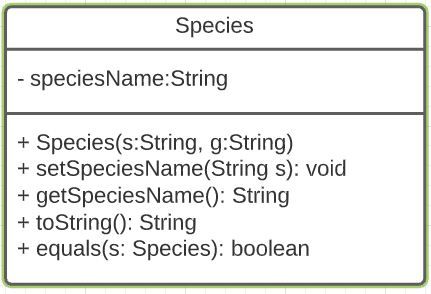
**Inheritance Forum Answers**

**Question1**

1. It is a parent-child relationship, with Genus being the parent and Species being the child
2. There is no relationship between Species and Specimen objects.
3. 
4. - We can reuse the code so we do not need to rewrite the same code for every child class we want to make. The child class can also access the fields and methods in the parent class.

- The readability of the code will be much better using inheritance (the parent-child relationship between the 3 classes) and the code is more structured.

1. i. Because each class has their own toString() method. Each toString() method is overridden so each class has their own unique toString() method and hence it won’t result in an error.

ii. The term is overriding.

**Question2**

1. Encapsulation or data hiding is the wrapping of data (variables) in data that acts as a single unit (such as classes in Java). The data attribute will be declared as private and the only way to access it is through the public getter and setter methods (which acts as the access points). Encapsulation acts as a “shield” to prevent data from being accessed outside of the class.
2. - Protects unwanted access by clients

- Easier to maintain the code and makes the codes easier to understand

1. getName(), getCage(), getTOA(),
2. name, cageNumber, toa
3. Located at Genus class.
4. Advantage: It will be a more structured code because at a taxonomy level, specimen is a subspecies. The specimen object could also access the fields and methods of the parent class (Species). Disadvantage: Some of the Specimen object’s methods and fields would be dependent on the parent class (Species), and it is difficult to make more customization on its fields and methods.

**Question3**

1. Add an instance variable inside the Specimen class called marking. Add another parameter for the marking inside the constructor of Specimen class. Add a getter and setter method for the marking.
2. Located at Specimen class.
3. Pseudocode for listSpecies

listSpecies (Specimen[] animals) {

LinkedList<String> allSpecies = new LinkedList <String>

for (each animal in Specimen[]){

if (animal's species has not existed)

insert animal's species to allSpecies

return allSpecies

}

**Question4**

1. We can perform a number of operations on abstract data types without knowing how the operation works.
2. Located at Specimen class.
3. Located at Species class.
4. Located at Species class