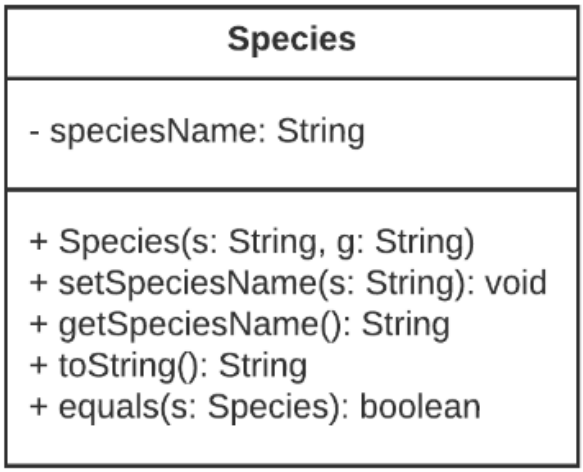
Question Set 1

a. The relationship between Genus and Species is inheritance as the Species subclass inherits from Genus, its superclass.
b. The relationship between Species and Specimen is composition as the Specimen class will not be able to function without the Species class.
c.



d. The inheritance relationship between the Genus and Species class allows programmers to reuse code that is present in the Genus class to be used in the Species class without needing to be rewritten in the Species class. This promotes reusability as there won't be duplicates of the same code in the two classes. The composition relationship between the Species and Specimen class allows the programmer to make changes internally in the code without worrying about compatibility and external side effects.
e.
    i. The toString() method doesn't cause an error because it overrides the toString() method in the Genus class with the method in the Species class.
    ii. The term of this property is dynamic polymorphism (overriding).

Question Set 2

a. Encapsulation in Java is the action of wrapping the variables and code acting on the methods together as a single unit. The variables are hidden from other classes and can only be accessed through its current class.
b. 1. The fields of a class can be changed to read-only or write-only.

2. Each class is able to maintain what is stored in its fields.

c. An accessor method in the Specimen class is the getName() method.
d. An instance method in the Specimen class is the name method.

e.

Java > OOP Week 9 Forum > Genus.java > ...

```java
public class Genus {
    private String genusName;
    public Genus(String g) {
        this.genusName = g;
    }

    public String getGenusName() { return genusName; }
    public String toString() { return "This animal belongs in the " + genusName + " Genus"; }
}
```

f.

g. Advantage: Methods in the Species class are inherited to the Specimen class which allows the methods to be reused without the method itself being rewritten.

h. Disadvantage: The Specimen class acts very similarly to the methods inherited from the Species class and the behavior of the Specimen classes are fairly limited to the methods created in the Species class.

Question Set 3

a. The changes needed in order to add a description of each animal's markings to the program is to create a private variable which aims to allow the markings of a specimen to be recorded. Furthermore, accessor and mutator methods can also be created to accompany the new variable.

b.

```java
public countSpecimens( Specimen[] animals, Species s ) {
    int i = 0;
    for (Specimen j : animals) {
        if (j.getTOA().equals(s)){
            i++;
        }
    }
    return i;
}
```

c. listSpecies(Specimen[] animals)

LinkedList<Specimen> listOfSpecies = new LinkedList<Specimen>();
for each animal in animals {
if animal.getTOA() is not in animals
listOfSpecies.add(animal.getTOA())
return animals

Question Set 4

a. 1. Provides essential information while hiding the details that make the essential information

2. Data types that are being operated on will not be able to know how they are being implemented

b.

```
public LinkedList makeList( Specimen[] animals ) {
// insert your code here
    LinkedList<Specimen> listOfSpecimen = new LinkedList<Specimen>();
    for (Specimen animal : animals) {
        listOfSpecimen.add(animal);
    }
    return listOfSpecimen;
}
```

c.

```
public makeSpeciesList( LinkedList animals ) {
    LinkedList<Species> listofSpecies = new LinkedList<Species>();
    for (Specimen animal : animals) {
        listofSpecies.add(animal.getTOA());
    }
    return listofSpecies;
}
```

d.

```
public makeSpeciesListUnique( LinkedList allSpecies ) {
    LinkedList<Species> listofSpeciesUnique = new LinkedList<Species>();
    for (Species species : allSpecies) {
        if (!listofSpeciesUnique.contains(species)) {
            listofSpeciesUnique.add(species);
        }
    }
    return listofSpeciesUnique;
}
```