1a. Inheritance, genus is the superclass while species is the subclass

1b. Aggregation, the specimen class cannot exist if the species class does not exist

|  |
| --- |
| **Species** |
| -speciesName: String |
| +Species(s: string, g: string)  +setSpeciesName(s: String):void  +getSpeciesName():string  +toString():string  +equals(s: species):boolean |

1d. -Easier to maintain data when a new species is found

-Easier to classify data as every species will have its own specimens

1e. Because there is another toString method on the class species

1e.(ii) Overriding

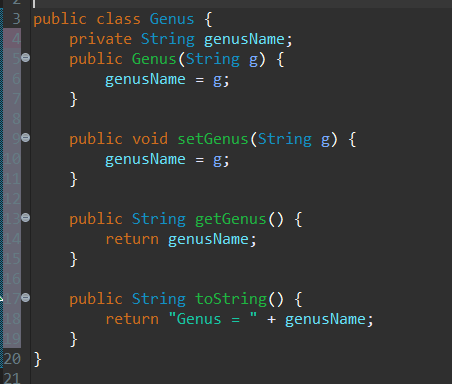
2a. It is part of the OOP concept whereby class variables are declared as private and accessing them requires the use of setter and getters

2b. -Ensures data entered is always valid

-Prevents the user from accessing information that they are not supposed to know

2c. getName()

2d. cageNumber

2e. 

2f.Advantage = It will be easier to identify when a new species is found when a certain specimen does not matches any of the currently known ones

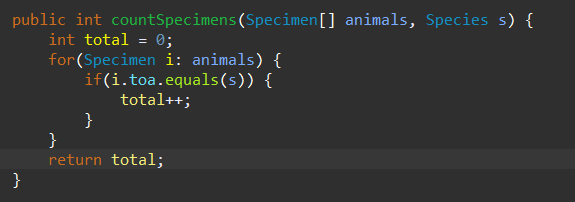
Disadvantage = It might take more time to fully comprehend the code for a new user

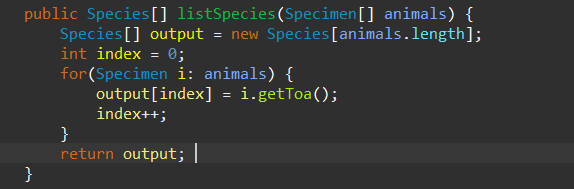
3a. -Create 2 new array for storing each specimen and it’s respective description

-When a new animal is added, add the new data to both arrays

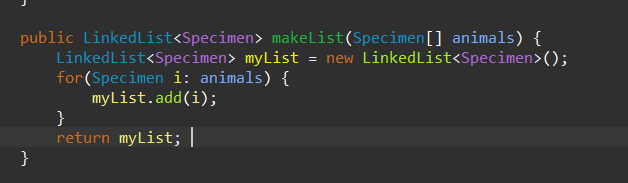
-This way, each animal’s description and name will have the same index

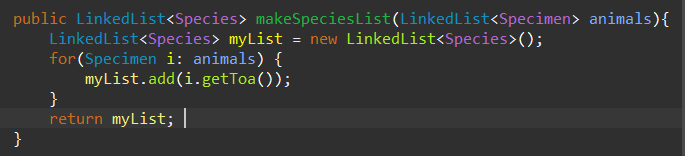
-Using a for loop, loop through each index of both arrays

3b. 

3c. 

4a. An abstract data type is a type for objects with methods and variables. It however does not explicitly state how each method functions and thus gives flexibility for the user. It generally acts as a building block for other data types.

4b. 

4c. 

4d. 