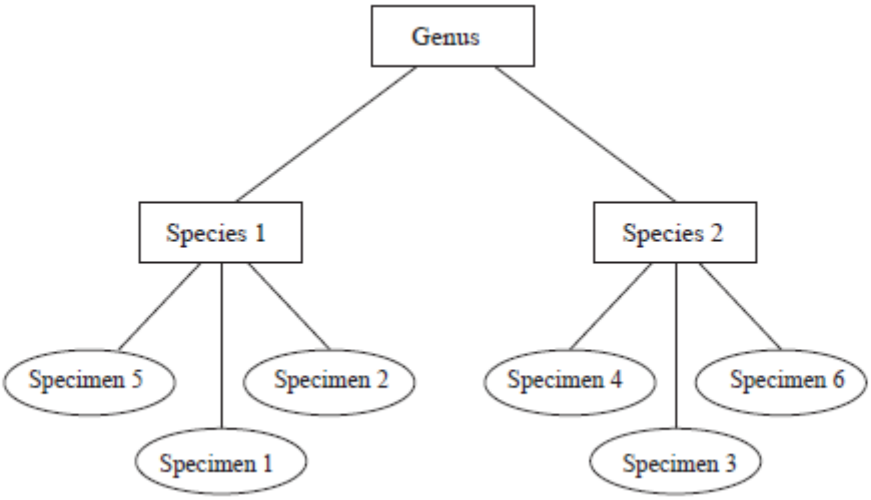
Inheritance Exercise

A large zoo has a collection of many individual animals of many different species. A computer program is being developed to keep track of all of the animals in the collection.

Because there are so many different kinds of species in the collection, and each species has some unique characteristics and some characteristics in common with other species, it was decided that the computer program would contain objects that correspond to different levels of

the taxonomy used by biologists to classify all life forms. A genus is composed of a group of species that have similar common characteristics, as shown in the diagram.



A separate object, `Specimen`, is used to represent each individual animal in the zoo.

The following code implements the `Species` and `Specimen` objects:

```
public class Species extends Genus
{
private String speciesName;
public Species( String s, String g )
{
super(g);
setSpeciesName(s);
}
public void setSpeciesName(String s){ speciesName = s; }
public String getSpeciesName(){ return speciesName; }
public String toString()
{
return "Species: " + getGenusName() + " " + speciesName;
}
public boolean equals(Species s)
{
return speciesName.equals(s.getSpeciesName());
}
}
```

```
public class Specimen
{
private String name;
private int cageNumber;
private Species toa; // "Type Of Animal"
public Specimen( String a, int c, Species s)
{
setName(a);
setCage(c);
setTOA(s);
}
public void setName(String a){ name = a; }
public void setCage(int c){ cageNumber = c; }
public void setTOA(Species s){ toa = s; }
public String getName(){ return name; }
public int getCage(){ return cageNumber; }
public Species getTOA(){ return toa; }
public String toString()
{
return name + " is a " + toa + " in cage " + cageNumber;
}
}
```

Question Set 1

(a) State the relationship between the `Genus` and `Species` objects. *[1]*

(b) State the relationship between the `Species` and `Specimen` objects. *[1]*

(c) Construct the unified modelling language (UML) diagram for the `Species` object. *[4]*

(d) Outline **two** ways in which the programming team can benefit from the way the relationships between the three objects, `Specimen`, `Species` and `Genus`, have been represented in the code. *[4]*

(e) The `Genus` class implements a `toString()` method that produces an output string that is different from the one produced by the `toString()` method in the `Species` class.

Consider the following code fragment:

```
Species human = new Species ("homo", "sapiens");
System.out.println(human.toString() );
```

(i) Outline why calling the `toString()` method in this code does not cause an error. *[2]*

(ii) Identify the term for this property. *[1]*

Question Set 2

(a) Define the term *encapsulation*. *[1]*

(b) Outline **two** benefits provided by encapsulation. *[4]*

(c) Identify an accessor method in the `Specimen` class. *[1]*

(d) Identify an instance variable in the `Specimen` class. *[1]*

(e) Construct code for the `Genus` object including a constructor, accessor methods and a `toString()` method. *[3]*

The `Specimen` object could have been designed as a sub-class of the `Species` object.

(f) Outline **one** advantage and **one** disadvantage of having the `Specimen` object as a sub-class of the `Species` object. *[4]*

Question Set 3

(a) Outline the changes that would be needed in order to add a description of each animal's individual markings to the program. *[4]*

An array is used to store the `Specimen` objects corresponding to the animals in the zoo.

(b) Construct a method `countSpecimens( Specimen[] animals, Species s )` that will output the number of specimens of the given species in the zoo. *[8]*

(c) Construct an algorithm in pseudocode for `listSpecies( Specimen[] animals )`, which will generate a list of the different species in the zoo. *[6]*

For some operations, it will be more convenient to use a linked list to hold the `Specimen` objects corresponding to the animals in the zoo. In answering this question, you should use the `LinkedList` class and may use the following methods.

```
addHead()
getHead()
addTail()
getTail()
getNext()
insert()
delete()
isEmpty()
isFull()
```

Question Set 4

(a) Identify the features of an abstract data type (ADT). *[2]*

(b) Construct a method, `makeList()`, that builds a linked list containing the `Specimen` objects from an input array.

```
LinkedList makeList( Specimen[] animals )
{
// insert your code here
}
```
[4]

(c) Construct a method, `makeSpeciesList( LinkedList animals )`, that will return a linked list of `Species` objects, one for each animal specimen present in the zoo.

Note that the list returned will contain duplicate `Species` objects if the zoo has more than one specimen of a given species. *[6]*

(d) Construct a method, `makeSpeciesListUnique( LinkedList allSpecies )`, that will take a linked list of `Species` objects as described in (c) and produce a new linked list that contains only one `Species` object for each species in the zoo. *[8]*

End of Exercise