CT3535 Assignment 4

Source Code:

Animal.java:

```
import java.io.Serializable;

public abstract class Animal implements Serializable, Comparable<Animal>{

	//Serial UID
	private static final long serialVersionUID = 1L;


	//initialising variable
	private String FirstName;
	private String LastName;
	private int Size;


	//constructor
	public Animal(String name, String surname, int size) {
		this.FirstName = name;
		this.LastName = surname;
		this.Size = size;
	}

	//compareTo Function, deal with it later
	public int compareTo(Animal A) {
		return this.getFirstName().compareTo(A.getFirstName());
	}

	//accessors & mutators
	public void setFirstName(String name) {
		this.FirstName = name;
	}

	public String getFirstName() {
		return FirstName;
	}

	public void setLastName(String name) {
		this.LastName = name;
	}

	public String getLastName() {
		return LastName;
	}

	public int getSize() {
		return Size;
	}

	//abstract sound method
	public abstract String sound();

	//toString method
	public String toString() {
		return FirstName + " " + LastName + ", " + Size;
	}
}
```

Dog.java:

```
public class Dog extends Animal{
    private static final long serialVersionUID = 1L;
    public Dog(String name, String surname, int size) {
            super(name, surname, size);
    }

    public String sound() {
        return "Woof!";
    }
}
```

Cat.java:

```
public class Cat extends Animal{
    private static final long serialVersionUID = 1L;
    public Cat(String name, String surname, int size) {
        super(name, surname, size);
    }

    public String sound() {
        return "Meow!";
    }
}
```

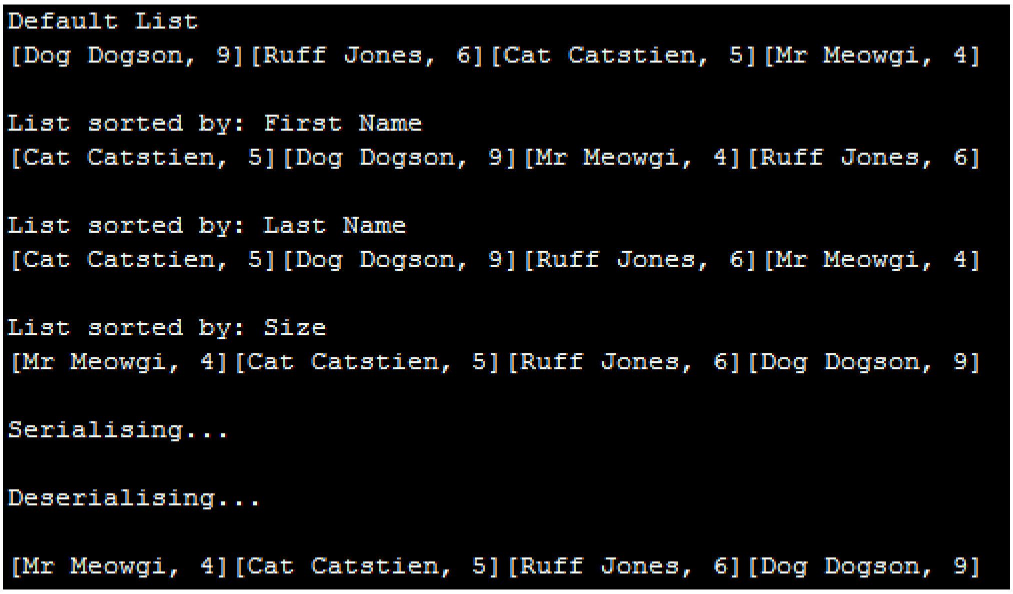
AnimalTester.java:

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.Collections;
import java.util.Comparator;
import java.util.LinkedList;
import java.util.List;

public class AnimalTester{

    public static void main(String[] args) {
        //lastNameCompare inner class for sorting list by lastName
        class lastNameCompare implements Comparator<Animal>{
            public int compare(Animal one, Animal two) {
                return
one.getLastName().compareTo(two.getLastName());
            }
        }

        //sizeCompare inner class for sorting list by size
        class sizeCompare implements Comparator<Animal>{
            public int compare(Animal one, Animal two) {
                Integer i1 = one.getSize();
                Integer i2 = two.getSize();
                return i1.compareTo(i2);
            }
        }

        //Initialising inner classes
        lastNameCompare lastNameComp = new lastNameCompare();
        sizeCompare sizeCompare = new sizeCompare();
```

```
//Dogs & Cats Variables
Dog DogA;
Dog DogB;
Cat CatA;
Cat CatB;

//LinkedList variable
List<Animal> AnimalHouse= new LinkedList<>();

//Creating instances & adding instances to LinkedList
DogA = new Dog("Dog", "Dogson", 9);
AnimalHouse.add(DogA);
DogB = new Dog("Ruff", "Jones", 6);
AnimalHouse.add(DogB);

CatA = new Cat("Cat", "Catstien", 5);
AnimalHouse.add(CatA);
CatB = new Cat("Mr", "Meowgi", 4);
AnimalHouse.add(CatB);

//printing out linkedList
System.out.println("Default List");
for (Animal element:AnimalHouse) {
      System.out.print("[" + element.toString() + "]");
}

System.out.println("\n");

//Collections.sort() by first name
Collections.sort(AnimalHouse);

//printing out sorted list by first name
System.out.println("List sorted by: First Name");
for (Animal element:AnimalHouse) {
      System.out.print("[" + element.toString() + "]");
}
System.out.println("\n");

//Collections.sort() by last name
Collections.sort(AnimalHouse, lastNameComp);

//printing out sorted list by last name
System.out.println("List sorted by: Last Name");
for (Animal element:AnimalHouse) {
      System.out.print("[" + element.toString() + "]");
}
System.out.println("\n");

//Collections.sort() by size
Collections.sort(AnimalHouse, sizeCompare);

//printing out sorted list by size
System.out.println("List sorted by: Size");
for (Animal element:AnimalHouse) {
      System.out.print("[" + element.toString() + "]");
}
System.out.println("\n");


//Calling Serialisation & Deserialisation
serialise(AnimalHouse);
```

```
            deserialise(AnimalHouse);
      }

            //Serialisation & Deserialisation
            public static void serialise(List<Animal> list){
            System.out.println("Serialising... \n");
              try {
                  FileOutputStream fileStream = new
FileOutputStream("AnimalHouse.dat");
                  ObjectOutputStream os = new
ObjectOutputStream(fileStream);
                  os.writeObject(list);

                  os.close();
            }catch (Exception e) {
                e.printStackTrace();
                  }
      }

            @SuppressWarnings("unchecked")
            public static void deserialise(List<Animal> list) {
                  System.out.println("Deserialising... \n");
                  try {
                        FileInputStream fileStream = new
FileInputStream("AnimalHouse.dat");
                        ObjectInputStream os = new
ObjectInputStream(fileStream);
                        List<Animal> ani = (List<Animal>)os.readObject();
                        for (Animal element: ani) {
                              System.out.print("[" + element.toString() +
"]");
                        }

                        os.close();
                  }catch (Exception e) {
                        e.printStackTrace();
                        }
      }
}
```

Screenshot:



```
Default List
[Dog Dogson, 9][Ruff Jones, 6][Cat Catstien, 5][Mr Meowgi, 4]

List sorted by: First Name
[Cat Catstien, 5][Dog Dogson, 9][Mr Meowgi, 4][Ruff Jones, 6]

List sorted by: Last Name
[Cat Catstien, 5][Dog Dogson, 9][Ruff Jones, 6][Mr Meowgi, 4]

List sorted by: Size
[Mr Meowgi, 4][Cat Catstien, 5][Ruff Jones, 6][Dog Dogson, 9]

Serialising...

Deserialising...

[Mr Meowgi, 4][Cat Catstien, 5][Ruff Jones, 6][Dog Dogson, 9]
```