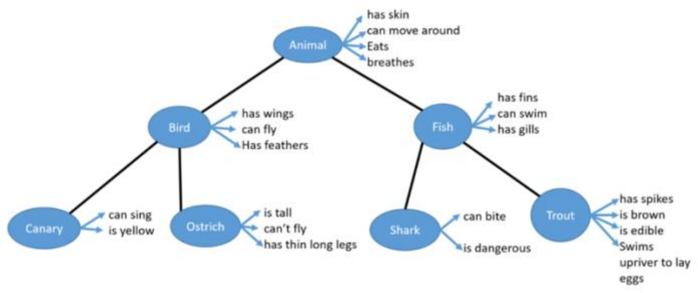
Assignment 3
Oisin Mc Laughlin – 22441106

Question 1 – Analysis

I will require 8 classes for this assignment; Animal, Bird, Fish, Canary, Ostrich, Shark and Trout. The focus of this assignment is inheritance which means I will be reusing a lot of code. I will be having a sort of hierarchy of classes to do this. Animal, Bird and Fish will be abstract classes so I won't be creating objects of them and just reusing the code I write in them in the other classes. Bird and Fish will be subclasses of Animal. Canary and Ostrich will be subclasses of Bird. Shark and Trout will be subclasses of Fish. A lot of the code is already provided for Animal, Bird and Canary but I will make small changes such as return types as they might throw me errors later on.



This is how the hierarchy will be laid out. Each class will have attributes that every other class has which I will define in the super classes such as Animal having; has skin, can move around, eats and breathes. Bird having; has wings, can fly and has feathers and Fish having; has fins, can swim and has gills. Canary, Ostrich, Shark and Trout also have unique attributes which I will define in those classes such as Canary having can sing and is yellow, Ostrich having; is tall, can't fly and has thin long legs. Shark having can bite and is dangerous. Trout having; has spikes, is brown, is edible and swims upriver to lay eggs. Most of these values are Booleans but there is some values that are Strings. Each of these subclasses/ leaf classes also will have a toString() method which will display each of the attributes of the animal including their unique ones which I will write up by looking at the canary example and adding some code to it. Each class will also have an equals method where I will compare each of the animals by using instanceof to make an input object to compare it to the class attributes using == for Booleans and .equals() for strings. Ostrichs cannot fly and I need to check the boolean value of flies in the move method for bird and display the appropriate output depending if it is true or false.

For the first test in AnimalTest I will simply create an array of size 4 and create an object for each animal at positions 0 to 3 and give each of them a name. I will loop through the array using .length to know when to stop and print out each of the animals toString() method displaying values for each animal.
For the second test I will also loop through the same animal array but this time I will have a nested for loop to compare the animal at the current position to each of the animals in the array using the .equals() method.

Question 2 – Coding

Canary:

```
public class Canary extends Bird
{

   //String name; // the name of this Canary

   /**
    * Constructor for objects of class Canary
    */
   public Canary(String name)
   {
      super(); // call the constructor of the superclass Bird
      this.name = name;
      colour = "yellow"; // this overrides the value inherited from Bird

   }

   /**
    * Sing method overrides the sing method
    * inherited from superclass Bird
    */
   @Override // good programming practice to use @Override to denote overridden
methods
   public String sing(){
      String singing = "tweet tweet tweet";
      return singing;
   }

   /**
    * toString method rfeturns a String representation of the bird
    * What superclass has Canary inherited this method from?
    */
   @Override
   public String toString(){
      //Printing out each of the attributes of the class including the unique ones
      String strng ="";
      strng+= "\n\n\n";
      strng+= "-Canary\n";
      strng+= "-Name:\n";
      strng+= name;
      strng+= "\n-Colour:\n";
      strng+= colour;
      strng+= "\n-Skin?\n";
      strng+= hasSkin();
      strng+= "\n-Breath?\n";
```

```
        strng+= breathes();
        strng+= "\n-Eats?\n";
        strng+= eats();

        strng+= "\n-Feathers?\n";
        strng+= hasFeathers();
        strng+= "\n-Wings?\n";
        strng+= hasWings();
        strng+= "\n-Flies?\n";
        strng+= flies();
        strng+= "\n-Singing:\n";
        strng+= sing();
        strng+= "\n";
        strng+= move(5);

        return strng;
    }


    /**
     * equals method defines how equality is defined between
     * the instances of the Canary class
     * param Object
     * return true or false depending on whether the input object is
     * equal to this Canary object
     */

    @Override
    public boolean equals(Object obj) {
        //Checking for null object
        if (obj == null) {
            System.out.println("NULL Object\n");
            return false;
        }

        //If object is instance of canary
        if (obj instanceof Canary) {
            //Set input object to instance of canary
            Canary canary = (Canary) obj;

            //Comparing attributes of the class
            if (this.getName().equals(canary.getName()) &&
this.getColour().equals(canary.getColour())) {
                return true;
            }
        }
        return false;
    }
}
```

Ostrich:

```
public class Ostrich extends Bird
{
    //String name;
    boolean isTall;
    String legs;

    /**
     * Constructor for objects of class Ostrich
     */
    public Ostrich(String name)
    {
        super();
        this.name = name;
        flies = false;
        //colour = "brown";
        //distance = 0;

        isTall = true;
        legs = "thin long legs";
    }

    public boolean isTall() {
        return isTall;
    }
    public String legs() {
        return legs;
    }

    @Override
    public String toString() {
        //Printing out each of the attributes of the class including the unique ones
        String strng ="";
        strng+= "\n\n\n";
        strng+= "Ostrich\n";
        strng+= "-Name:\n";
        strng+= name;
        strng+= "\n-Colour:\n";
        strng+= colour;
        strng+= "\n-Skin?\n";
        strng+= hasSkin();
        strng+= "\n-Breath?\n";
        strng+= breathes();
        strng+= "\n-Eats?\n";
        strng+= eats();

        strng+= "\n-Feathers?\n";
        strng+= hasFeathers();
        strng+= "\n-Wings?\n";
        strng+= hasWings();
        strng+= "\n-Flies?\n";
        strng+= flies();
        strng+= "\n-Singing:\n";
        strng+= sing();
        strng+= "\n";
        strng+= move(10);

        strng+= "\n-Tall?\n";
        strng+= isTall();
        strng+="\n-Legs?\n";
        strng+= legs();

        return strng;
    }

    @Override
    public boolean equals(Object obj) {
        //Im running out of time and have to go work, I explained the equals class in Canary
object, thanks.
        if (obj == null) {
            System.out.println("NULL Object\n");
            return false;
        }

        if (obj instanceof Ostrich) {
            Ostrich ostrich = (Ostrich) obj;

            if (this.getName().equals(ostrich.getName()) &&
this.getColour().equals(ostrich.getColour()) && this.isTall() == ostrich.isTall()) {
                return true;
            }
        }

        return false;
    }
}
```

Fish:

```
public abstract class Fish extends Animal
{
    boolean hasFins;
    boolean canSwim;
    boolean hasGills;

    /**
     * Constructor for objects of class Fish
     */
    public Fish()
    {
        super();

        colour = "white";

        hasFins = true;
        canSwim = true;
        hasGills = true;
    }

    @Override
    public String move(int distance) {
        String moving = "I swim " + distance + " metres \n";
        return moving;
    }

    public boolean hasFins() {
        return hasFins;
    }
    public boolean canSwim() {
        return canSwim;
    }
    public boolean hasGills() {
        return hasGills;
    }
}
```

Shark:

```
public class Shark extends Fish
{
    //String name;
    boolean canBite;
    boolean isDangerous;
```

```java
/**
 * Constructor for objects of class Shark
 */
public Shark(String name)
{
   super();
   this.name = name;

   canBite = true;
   isDangerous = true;
}

public boolean canBite() {
   return canBite;
}
public boolean isDangerous() {
   return isDangerous;
}

@Override
public String toString() {
   //Printing out each of the attributes of the class including the unique ones
   String strng ="";
   strng+= "\n\n\n";
   strng+= "Shark\n";
   strng+= "-Name:\n";
   strng+= name;
   strng+= "\n-Colour:\n";
   strng+= colour;
   strng+= "\n-Skin?\n";
   strng+= hasSkin();
   strng+= "\n-Breath?\n";
   strng+= breathes();
   strng+= "\n-Eats?\n";
   strng+= eats();

   strng+= "\n-Fins?\n";
   strng+= hasFins();
   strng+= "\n-Swims?\n";
   strng+= canSwim();
   strng+= "\n-Gills?\n";
   strng+= hasGills();
   strng+= "\n";
   strng+= move(10);

   strng+= "\n-Bite?\n";
```

```
        strng+= canBite();
        strng+="\n-Dangerous?\n";
        strng+= isDangerous();

        return strng;
    }

    @Override
    public boolean equals(Object obj) {
        //Im running out of time and have to go work, I explained the equals class in Canary
object, thanks.
        if (obj == null) {
            System.out.println("NULL Object\n");
            return false;
        }

        if (obj instanceof Shark) {
            Shark shark = (Shark) obj;

            if (this.getName().equals(shark.getName()) &&
this.getColour().equals(shark.getColour()) && this.canBite() == shark.canBite() &&
this.isDangerous() == shark.isDangerous()); {
                return true;
            }
        }

        return false;
    }
}
```

Trout:

```
public class Trout extends Fish
{
    //String name;
    boolean hasSpikes;
    boolean isEdible;
    String eggs;

    /**
     * Constructor for objects of class Trout
     */
    public Trout(String name)
    {
        super();
        this.name = name;
    colour = "brown";

    hasSpikes = true;
    isEdible = true;
    eggs = "Swim upriver to lay eggs";
 }

 public boolean hasSpikes() {
    return hasSpikes;
 }
 public boolean isEdible() {
    return isEdible;
 }
 public String laysEggs() {
    return eggs;
 }

 @Override
 public String toString() {
    //Printing out each of the attributes of the class including the unique ones
    String strng ="";
    strng+= "\n\n\n";
    strng+= "Trout\n";
    strng+= "-Name:\n";
    strng+= name;
    strng+= "\n-Colour:\n";
    strng+= colour;
    strng+= "\n-Skin?\n";
    strng+= hasSkin();
    strng+= "\n-Breath?\n";
    strng+= breathes();
    strng+= "\n-Eats?\n";
    strng+= eats();

    strng+= "\n-Fins?\n";
    strng+= hasFins();
    strng+= "\n-Swims?\n";
    strng+= canSwim();
    strng+= "\n-Gills?\n";
    strng+= hasGills();
    strng+= "\n";
    strng+= move(10);

    strng+= "\n-Spikes?\n";
    strng+= hasSpikes();
    strng+= "\n-Spikes?\n";
    strng+= hasSpikes();
```

```
        strng+= "\n-Where eggs?\n";
        strng+= laysEggs();

        return strng;
    }

    @Override
    public boolean equals(Object obj) {
        //Im running out of time and have to go work, I explained the equals class in Canary
object, thanks.
        if (obj == null) {
            System.out.println("NULL Object\n");
            return false;
        }

        if (obj instanceof Trout) {
            Trout trout = (Trout) obj;

            if (this.getName().equals(trout.getName()) &&
this.getColour().equals(trout.getColour()) && this.hasSpikes() == trout.hasSpikes() &&
this.isEdible() == trout.isEdible() && this.laysEggs().equals(trout.laysEggs())); {
                return true;
            }
        }

        return false;
    }
}
```

AnimalTest:

```
public class AnimalTest
{
    /**
     * Constructor for objects of class AnimalTest
     */
    public AnimalTest()
    {
    }

    public static void main(String[] args) {
        //Creating animal test objects and calling test methods.
        AnimalTest test = new AnimalTest();
        test.test1();
        test.test2();
    }
    public void test1() {
        System.out.println("\n\n\nTest 1\n\n");

        //Animal array creating objects at each position of array
        Animal[] animals = new Animal[4];
        animals[0] = new Canary("Oisin");
        animals[1] = new Ostrich("Ciaran");
        animals[2] = new Shark("Paul");
        animals[3] = new Trout("Daniel");

        //Looping through while i is less than animals array length and printing the toString of
each object in array.
        for (int i = 0; i < animals.length; i++) {
            System.out.println(animals[i]);
            System.out.println("\n");
        }
    }

    public void test2() {
        System.out.println("\n\n\nTest 2\n\n");

        Animal[] animals = new Animal[4];
        animals[0] = new Canary("Oisin");
        animals[1] = new Ostrich("Ciaran");
        animals[2] = new Trout("Daniel");
        animals[3] = new Shark("Paul");

        //Looping through animal array again but nested for loop to compare that animal to
each of the animals in the array.
        for (int i = 0; i < animals.length; i++) {
            for (int j = 0; j < animals.length; j++) {
                System.out.println("Comparing " + animals[i].getName() + " to " +
animals[j].getName());
                System.out.println(animals[i].equals(animals[j]));
            }
        }
    }
}
```

Animal (because I made small changes):

```
public abstract class Animal
{
    boolean hasSkin;
    boolean breathes;
```

```
//boolean canMove;
String colour;
String eats;
String name;

/**
 * Constructor for objects of class Animal
 */
public Animal()
{
   breathes = true; //all the subclasses of Animal inherit this property and value
   hasSkin = true; // all the subclasses of Animal inherit this property and value
   //canMove = true;
   colour = "grey"; //all the subclasses of Animal inherit this property and value
   eats = "food";
}
/**
 * move method
 * param int distance - the distance the Animal should move
 * All subclasses inherit this method
 */
public String move(int distance){
   String moving = "I move " + distance + " metres \n";
   return moving;
}

/**
 * getter method for colour field
 * All subclasses inherit this method
 */
public String getColour(){
   return colour;
}

 /**
 * 'getter' method for haSkin field
 * All subclasses inherit this method
 */

public boolean hasSkin(){
   return hasSkin;
}
public boolean breathes() {
   return breathes;
}
public String eats() {
   return eats;
```

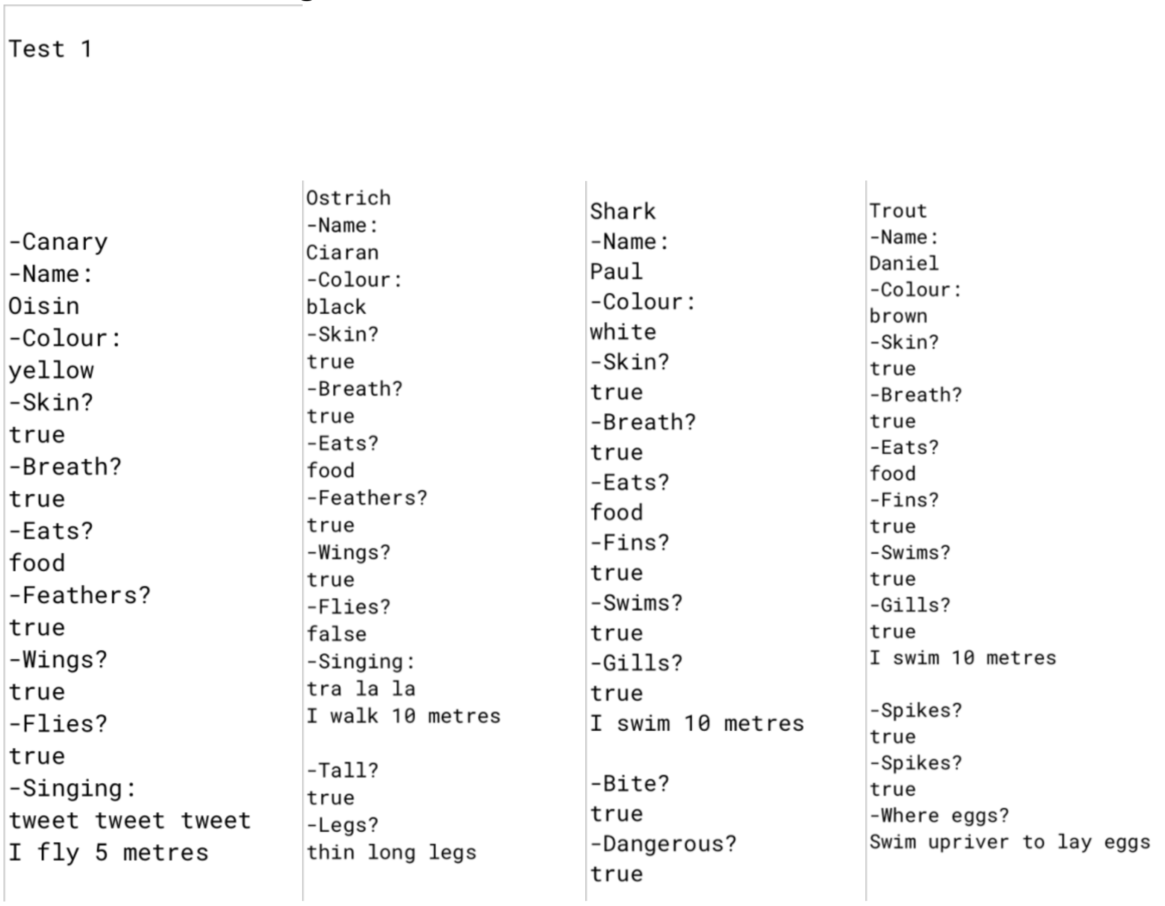
```
    }
    public String getName() {
        return name;
    }
}
```

Bird (because I made small changes):

```
public abstract class Bird extends Animal
{
    //instance variables (fields)
    boolean hasFeathers;
    boolean hasWings;
    boolean flies;

    /**
     * Constructor for objects of class Bird
     */
    public Bird()
    {
        super(); //calls the constructor of its superclass  - Animal
        colour = "black"; //overrides the value of colour inherited from Animal
        hasFeathers = true; //all the subclasses of Bird inherit this property and value
        hasWings = true; //all the subclasses of Bird inherit this property and value
        flies = true; //all the subclasses of Bird inherit this property and value
    }

    /**
     * move method overrides the move method
     * inherited from superclass Animal
     */
    @Override // good programming practice to use @Override to denote overridden
methods
    public String move(int distance){
        String moving;
        if (flies == true) {
            moving = "I fly " + distance + " metres \n";
        }
        else {
            moving = "I fly " + distance + " metres \n";
        }
        return moving;
    }

    /**
```

```
     * sing method that all birds have
     */
    public String sing(){
        String singing = "tra la la";
        return singing;
    }

    /**
     * 'getter' method for the hasWings field
     */
    public boolean hasWings(){
        return hasWings;
    }

    /**
     * 'getter' method for the hasFeathers field
     */
    public boolean hasFeathers(){
        return hasFeathers;
    }

    public boolean flies() {
        return flies;
    }
}
```

Question 3 – Testing



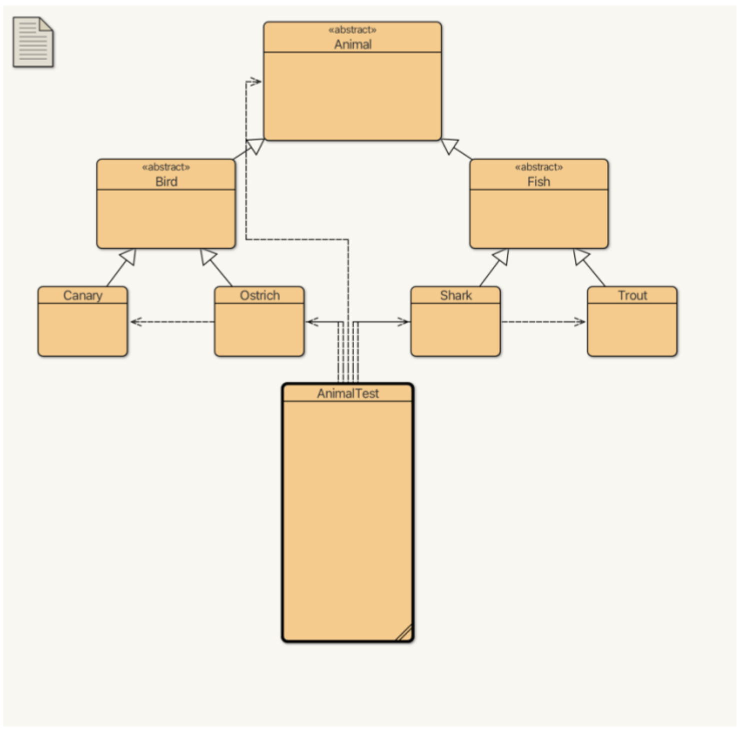
```
Test 1



-Canary
-Name:
Oisin
-Colour:
yellow
-Skin?
true
-Breath?
true
-Eats?
food
-Feathers?
true
-Wings?
true
-Flies?
true
-Singing:
tweet tweet tweet
I fly 5 metres
```

```
Ostrich
-Name:
Ciaran
-Colour:
black
-Skin?
true
-Breath?
true
-Eats?
food
-Feathers?
true
-Wings?
true
-Flies?
false
-Singing:
tra la la
I walk 10 metres

-Tall?
true
-Legs?
thin long legs
```

```
Shark
-Name:
Paul
-Colour:
white
-Skin?
true
-Breath?
true
-Eats?
food
-Fins?
true
-Swims?
true
-Gills?
true
I swim 10 metres

-Bite?
true
-Dangerous?
true
```

```
Trout
-Name:
Daniel
-Colour:
brown
-Skin?
true
-Breath?
true
-Eats?
food
-Fins?
true
-Swims?
true
-Gills?
true
I swim 10 metres

-Spikes?
true
-Spikes?
true
-Where eggs?
Swim upriver to lay eggs
```

```
Test 2


Comparing Oisin to Oisin
true
Comparing Oisin to Ciaran
false
Comparing Oisin to Daniel
false
Comparing Oisin to Paul
false
Comparing Ciaran to Oisin
false
Comparing Ciaran to Ciaran
true
Comparing Ciaran to Daniel
false
Comparing Ciaran to Paul
false
Comparing Daniel to Oisin
false
Comparing Daniel to Ciaran
false
Comparing Daniel to Daniel
true
Comparing Daniel to Paul
false
Comparing Paul to Oisin
false
Comparing Paul to Ciaran
false
Comparing Paul to Daniel
false
Comparing Paul to Paul
true
```



Each of the tests gave the output as expected but I may have changed test 2 to display more information but I currently don't have the time to do it due to having work clashing with the time I need to do that.