

CS101: Lab #20

Inheritance Part II

In this lab, we will build off what we learned in the last lab.

Our goal is to learn more about inheritance, specifically polymorphism and how to treat a set of subclasses uniformly based on the super class interface. We will need the same classes as we did for the last lab. So, copy all 7 classes from the prior lab project to your new lab project.

Now make the following changes:

- 1.) In the **Animal** class, add an input **String** parameter to the constructor called **type**, which we will use to identify the type of animal. Add the appropriate private member variable to copy the input parameter's value into and a public accessor method called **getType()**.
- 2.) In each of the **Dog**, **Cat**, **Bird**, **Fish**, and **Fox** classes, add the corresponding type of the animal to the call to the super class constructor. To be super clear by type, I mean "Dog" for the **Dog** class, etc...
- 3.) Back in the **Animal** class, make it an abstract class and make each of the following methods abstract: **numberOfLegs()**, **numberOfWings()**, and **breatheUnderwater()**.
- 4.) In the **Tester** class:
 - a. Remove all the code in the main method aside from the creation of the five animal objects.
 - b. Create and initialize a variable of type **ArrayList<Animal>**.
 - c. Add each of the five animals into the **ArrayList**.
 - d. Use the enhanced for loop syntax to iterate over the **ArrayList** and call the **PrintDetails** method for each animal in the **ArrayList**.
 - e. Create a method with the following signature:

```
public static void PrintDetails( Animal a )
```

That method should use the public methods available on the **Animal** class to determine if the animal it was passed has legs, wings or gills and in each case call **numberOfLegs()**, **numberOfWings()**, or **breatheUnderwater()** as appropriate. Also call **getSoundMade()**. You will need to use the **getType()** method we added in order to produce the sample output below exactly.

Once you have implemented your code correctly, running the tester class should produce this *exact* output:

```
A Dog has 4 legs.
A Dog makes the sound woof
A Cat has 4 legs.
A Cat makes the sound meow
A Bird has 2 legs.
A Bird has 2 wings.
A Bird makes the sound tweet
A Fish can breathe under water.
A Fish makes the sound blub
A Fox has 4 legs.
A Fox makes the sound Ring-ding-ding-ding-dingeringeding!
```