The purpose of this lab is to get more practice with setting up java classes. And to get practice in writing methods and using encapsulation.

**Part 1 – Java class**

**Note**: You can use the same java project and package within the project as you did last week – or set up a *new* package within the same project. Or set up a new project and package. It’s up to you!

Create a new class called Animal. Give it the following class members:

**5 attributes** - name, breed, number of legs, domestic (or not), colour; For now, DON’T make them as private.

**2 constructors**

* One constructor that sets up just the animal name;
* Another constructor that sets up all 5 attributes.
* Put a System.out.println statement into each of the constructors that prints out the object values in a readable way e.g. “This object is a dog that is white in colour and has 4 legs and is domestic”.. etc

Use a main method in a separate class (e.g. called Control) to instantiate the following objects, including:

A domestic 4-legged brown dog with called Spot

An animal called “Leo”;

… more of your own choice!

**PART 2 – Encapsulation - hide**

The idea of encapsulation is “hiding” things that shouldn’t be seen. E.g. When you use a coffee machine, you just use the external buttons – you don’t see the innards of the machine, and add the powder yourself etc. OO programming works in the same way. Let’s illustrate this:

In your “Main” method of your control class, try to print out the attribute values directly of any object you have created e.g.

System.out.println(*whateveryourobjectnameis*.name);

System.out.println(*whateveryourobjectnameis*.domestic);

System.out.println(*whateveryourobjectnameis*.numberOfLegs);

UNLESS your attributes in your Animal class have been marked private, you’ll be allowed to access these attributes of your object from another class, and to see their values. See if you can update them to new values. i.e. animal.name = “.. whatever”.

Now mark each attribute in your Animal class as private – e.g.

Eg. Private String name ;… The compiler will then prevent you from access the attributes directly from another class.

Marking the attributes as “private” in the class is the first step to *encapsulating* them.

**PART 3 – getters and setters methods to support Encapsulation**

Once you have your attributes all set to private from Part 2, you can then add methods to your class that let you set the values of the attributes and “get” the values.

**Setters**

* Setter methods allow you to change/set the value of an object attribute.
* Add a method public void setName (String name) that allows the name attribute to be reset.
* Add in more setters : setDomestic and setNumberOfLegs etc.. for all 5 attributes in your class.

From your main method, using an animal object you set up earlier – call your new setter methods to change each of the attribute values.

**Getters**

Getter methods allow you to retrieve the values of all the attributes.

* Add a method public String getName () that returns the current value of the animal name.
* Add in more getter methods : getDomestic and getColour etc.. for all attributes in your Animal class

From your main method, using an animal object you set up earlier – call your new getter methods to retrieve each of the attributes values.

**PART 4 – Overloading methods**

Overloading methods means having more than one method in a class with the same name.

In your animal class, add a method called makeNoise();

In it, just print out a noise… put in an if condition to do this. If the breed is a dog, print out a Bark, if it’s a cat, “miaow” etc.. for whatever breeds you have.

if (breed.equals(“dog”))

{

System.out.println(“woof woof”);

} else if ()

{

// some other condition;

} else if ()

{

// some condition;

etc

Now, add another makeNoise () method – but this times it takes in a String parameter “volume”. If volume is “loud”, print in uppercase. Note that you are allowed to have methods with the same name.

Call both methods on your animal object from the main method in the Control class.