The purpose of this lab is to get more practice with setting up java classes, particularly writing *methods* and using *encapsulation,* and using the Java API.

**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. Up to you…

**Part 1 – Java class – 0.3**

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

**3 attributes in your Pet class** - (1) name of the pet, (2) type (3) whether the pet is trained or not.

For now, **DON’T** mark the attributes as private for now.

**Add a constructor in your Pet class**  that sets up all 3 attributes.

**Add a toString() method in your Pet class**

As we did last week in the lab, add a toString() method to your class that returns a String that contains the object data in a readable way e.g. “This pet is called spot, is a retriever and it is true that spot is trained”

Create another java class (call it Control) and put a main method in it. In this “main” method, instantiate a Pet object (e.g. an untrained labrador called .. whatever).

From the main method, print out the object you have created (Using System.out.println(objectname).

Try accessing the attribute values from the main method:

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

What happens and why?

**PART 2 – Encapsulation –– 0.6**

Encapsulate the three attributes in the Pet class:

* Set each of them to private;
* Add getter and setters methods (e.g. public String getNam(), public void setName(String name)… etc

From your main method, try out your six getter and setter methods.

Use the getters in your toString() and setters in your Constructor.

**PART 3 – Using methods 0.8**

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

Public void makeNoise()

{

}

In it, just print out a noise… put in an if condition to do this. If the type is a dog, print out “bark”, if it’s a cat, “miaow” , else “not making any noise”.

Test your makeNoise() method by calling it for your Pet objects in the main method.

**Part 4 – Use the Java API - 1**

**As covered in the tutorial, the Java API is a library of “free” java classes that do common things that save you writing all your code from scratch.** The API is available on the Oracle website :

Exercise: For the API exercise, you don’t have to test the code, just write it out in an editor such as notepad++ or in your IDE in a main method  
Go the Java API at this link: https://docs.oracle.com/en/java/javase/22/docs/api/index.html

1. Find the **JTextfield** class   
   (Use the search bar on top right to help)
   1. What **module** and **package** is it in?
   2. What is the purpose of this class (find it’s definition in the API definition)
   3. How many “Constructors” does JTextfield have?
   4. Write the (2 lines of) code to instantiate :
      1. Instantiate JTextfield object , initialised with some sample text.
      2. Sets the horizontal alignment of the text to 3.
2. You’re writing a piece of code that compares two strings. Find the String class in the API. From examining the methods available in this class, write the two lines of code needed to compare two strings as follows:

* Create a String object containing “this is the first string”;
* Compare that string to the string “this is the first String1”;