

## Assignment 2: Object-Oriented Programming

In this assignment, you'll need to build and run a small Zoo in Lennoxville. All classes must be created in separate files.

### Animal (5)

First, start by building a class that describes an Animal at a Zoo. It should have one private instance variable for the name of the animal, and one for its hunger status (fed or hungry).

Add methods for setting and getting the hunger status variable, along with a getter for the name. Consider how these should be named for code clarity. For instance, using a method called `hungry()` to make the animal hungry could be used as a setter for the hunger field. The same logic could be applied to when it's being fed:

```
public void feed() {  
    this.fed = true;  
}
```

Furthermore, the getter for the fed variable could be named `isFed` as it is more descriptive about what it answers when compared to `getFed`. Keep this technique in mind for future class designs.

### Zoo (10)

Now we have the animals designed and ready for building a little Zoo!

Build a class called Zoo that contains an animal collection and a given address. Given that we'd like to increase the number of animals we have over time, we won't be able to use standard arrays. To have a dynamically-sized, resizable, or growable array, we will make use of the ArrayList class:

<https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/ArrayList.html>

To make a vector for the Animal collection, you can do the following:

```
ArrayList<Animal> animalCollection = new ArrayList<Animal>();  
animalCollection.add(new Animal("Chloe the Bear"));
```

Now let's get ready for some customers. Ensure that your Zoo can do the following:

1. Print opening hours for all Zoos (they all open and close at the same time 9AM-5PM).
2. Print the address of the Zoo.
3. Add an Animal to the Animal collection.
4. Feed, and set hunger for the Animals in the collection.
5. Print all Animals that are hungry in the Zoo.

With all these methods implemented we are ready to open to the public!

## Pre-opening Day (5)

You've realized that you should test your implementation before opening it to the public! This means you will act as the first customer to the Zoos you are building, and perform at least the following in your main method:

1. Enter your Zoos (one is at "122 Queen St.", and the other at "248 College St.").
2. In the Zoo on Queen street, add these four Animals to the collection:
  - a. Chloe the Bear
  - b. Pete the Parrot
  - c. Sylvestre the Snake
  - d. Manix the Monkey
3. Leave the Zoo on College street empty.
4. Print the opening hours for the Zoos, and make sure they all match (i.e. not unique to instances).
5. Do the same for the addresses but this time they shouldn't match.
6. Feed "Chloe the Bear" at the Zoo on Queen St., and without it getting hungry again, try to feed it again.
7. Feed the same Animal at the Zoo on College St.
8. Print the Animals that are hungry at each Zoo.
9. Play with "Chloe the Bear" to make it hungry.
10. Re-print the Animals that are hungry.

## Competition (10)

A competing zoo opened up next door on the same day as you. The Zoo has all the same features as yours currently has. Implement some feature(s) in your Zoo that will attract more people to yours! It should be created using 2 additional classes.

**Grading Criteria:**

Style/submission guidelines: [https://gmierzwinski.github.io/bishops/cs321/style\\_guidelines.html](https://gmierzwinski.github.io/bishops/cs321/style_guidelines.html)

<b>Comments, Formatting, &amp; Readability</b>	<b>5 Marks</b>
<b>Submission Guidelines</b>	<b>5 Marks</b>
<b>Parts 1-3</b>	<b>30 Marks</b> <b>See (X) above</b>
<b>Total</b>	<b>40 Marks</b>