CS5004-5005-SV – Lab 6 – June 11, 2020
Inheritance

The main goal of today's lab is to practice using inheritance and other topics discussed in recent lectures. In programming the following, please pay attention to: class hierarchy, *extends*, *super*, *static*, *final*, error checking, and overriding.

0. Create a new Class called **Bird**. Birds have these instance variables: double wingSpan, String color, String [] foodTypes. Create a no-argument constructor, getters, setters, toPrint method, and equals method for this class. Also create a private final variable for numWings, initialized to 2, with a public getter. Bird class should also have a final, static variable called allFoods. There is a constraint on the foodTypes instance variable that it be non-empty and that every String in the array must be included in allFoods. For simplicity, you may assume that all Birds will eat berries, but some favor small mammals, some prefer nectar, and others eat seeds and nuts.

1. Create a subclass of Bird called **Moa**. It turns out that Moas, which are now instinct, had no wings! How can you represent this without editing the parent class? Use super() in your constructor and demonstrate the correct behavior regarding the number of wings. Write new methods or override inherited methods, as needed, to incorporate this new/changed information.

2. Create a subclass of Bird called TalkingBird. It should have an int instance variable, vocab, for the number of words/phrases in its vocabulary, with getter and setter. It should restrict the number to be >0 and <100. Write the constructor in such a way that duplication of code for error checking is minimized. Override any methods or create new methods, as appropriate, to incorporate this new/changed information.

3. Create subclasses of TalkingBird called Mynah and Parrot. Mynahs are omnivorous. Be sure to represent that in the constructor and ensure this property is true of all instances, including even future subclasses of Mynah. Some parrots prey on animals such as snails or invertebrate larvae. Also, parrots have a favorite greeting. Users of this subclass are *not* allowed to change the greeting, and it must be one of "Hello!", "Howdy!", "Ay, Captain", and "'Sup?". Initialize the instance variable and check for errors.

4. Choose a new type of Bird you find on the WWW, and merge it into the hierarchy. It should have at least one new feature that is not part of Bird.

5. Create an array of instances of all *different types* of birds, and print it.
6. Create a main method that tests all the above and prints examples.
7. Returning to the **Date** example -- create a new Class called Assignment. An assignment consists of the Date Due, and the Body, a string representing the filename describing the assignment. Create two versions of a Copy constructor for Assignment. In one version, it should return a new *copy* of the Date field. In another version, it should simply return the Date field, like one might return an int or the String for the filename. Test these by creating a new assignment due tomorrow and then creating another assignment -- using the Copy Constructor on the first assignment. Use a setter change the filename to a new name. Then use the setter from the Date class to make the second assignment due 3 days later. Document your results, in each case, in line commentary. Turn in both versions.

Submit your answers to above items to Canvas for CS5005 *(due next Thursday by midnight). Only your .java files for the above need to be uploaded. Prompt submission encouraged, but late submissions are still better than not submitting.*

8. Work on any unfinished labs or CS5004 homework assignments. If finished, ***submit homeworks to Canvas CS5004***.