asgn01 Classes

Objectives

- Create a class
- Create class variables
- Create class methods
- Read class diagrams

The Assignment

You will learn OOP (Object Oriented Programming) using PHP in the class. One of the values of OOP is code reusability.

- Complete the bike challenge. I realize that he provides an answer.
- Complete the bird class exercise below.

Folder structure

```
web250
    asgn01
        bike-challenge
        bird-challenge
```

git

Inside your **web250** folder you will need to run the following commands just as you did for **asgn00**

You should not need to run `git init` since you initialized your repository for **asgn00**

```
git add .
git status
git commit -m"Created a Bike class"
```

1. Bike Challenge

Watch the LinkedIn/Learning series titled PHP: Object-Oriented Programming by Kevin Skoglund.

For this assignment, watch the Introduction and these chapters.

1. Introduction (Make sure to download the exercise material).
2. Overview and Project Setup
3. Object Basics

Complete the Challenge at the end of Object Basics. It will serve as a template for the assignment. Mr. Skoglund provides a solution but give it your best shot without looking at the answer and without wasting too much time. Or, watch his answer, and try to code the challenge without referencing it. Challenge yourself.

git

After finishing the Bike Challenge stage (add .) and commit your files again. It is this same series of commands as before.

UML - Unified Modelling Language

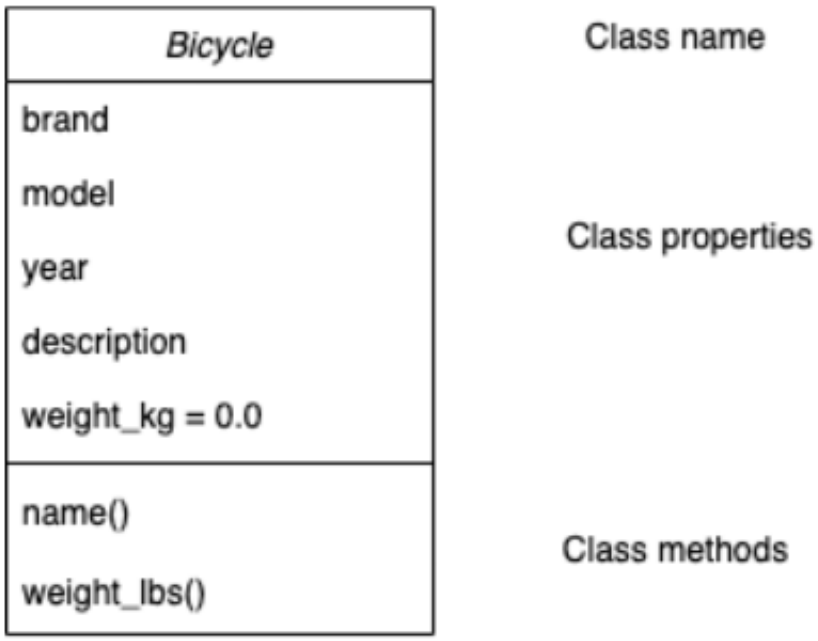
UML is a diagramming system comprised of many modeling techniques. We will use it for class diagrams just like the ones you saw in the OOD series. You can think of the class diagrams as a visual algorithm but it is closer to an ERD in my opinion. They are incredibly helpful in organizing your thoughts as your programs become more complex.

Here is a class diagram for the Challenge. Note that The class name is a singular noun, and the first letter is capitalized.

Use CamelCase if the class name is more than one word.

The properties (attributes) are located below the class name. Here you can include a default value.

The bottom section displays the methods (OOP speak), also called functions.



NOTE: The author's code is dated. He is using HTML4 and also uses underscores in his file names. You have a choice

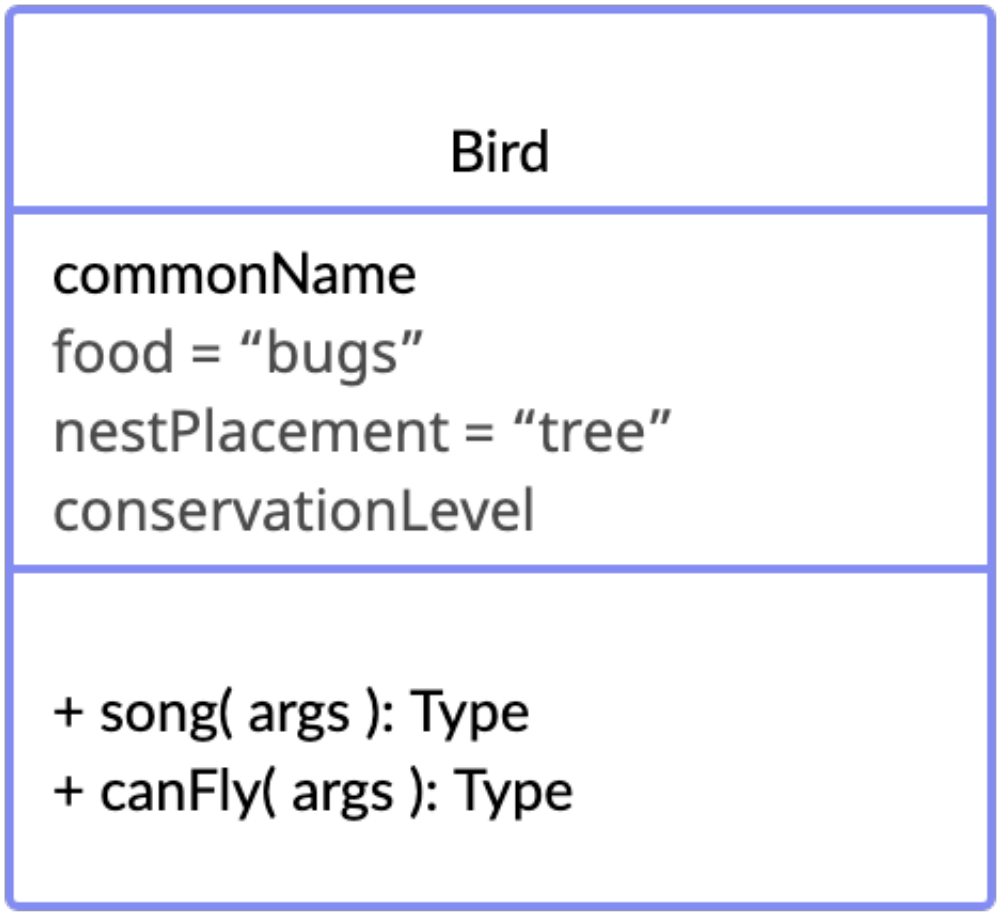
1. Continue to use his coding technique for his exercises
2. Change them to fit the coding standards listed in Moodle.

We will use the coding standards in listed in Moodle for the Bird Challenge later in this assignment.

2. Bird Challenge

Create a file named **index.php** Inside your **bird-challenge** folder.

Code the class from this diagram



(ignore the + signs for now). I have left in some default values but you do not have to use them.

Create an instance from the diagram

Create two bird instances. Name the instances `$bird1` and `$bird2`. Here is the data and menthod information.

git

After you create you Bird class, it is time to stage and commit again. Honestly, this is probably more frequent than you would usually commit, but it is good practice as you learn version control.

```
git add .
git status
git commit -m"Created a Bird class"
```

$bird1

```
Properties
-------------

commonName = Eastern Towhee
foodfood = seeds, fruits, insects, spiders
nestPlacementnestPlacement = Ground
conservationLevelconservationLevel = Low

Methods
-------------

song = drink-your-tea!
canFly = This bird can fly
```

$bird2

```
Properties
-------------

commonName = Indigo Bunting
food = small seeds, berries, buds, and insects
nestPlacement = roadsides, and railroad rights-of-wafields and on the edges
conservationLevel = Low

Methods
------------------------------------------------------

birdSongbirdSong == whatwhat!!
canFly = This bird can fly
```

You will need to display the content for both bird instances.

git

After finishing the Bike Challenge stage (add .) and commit your files again. It is this same series of commands as before.

GitHub

Now that everything is running correctly, it is time to push your local repo up to GitHub.

```
git add .
git status
git commit -m"Finished the Bike and Bird classes."
git push
```

Open your GitHub account to make sure everything pushed to your account. If it didn't, Git usually has informative messages on what went wrong.

Submit your GitHub address

Submit your WEB250 GitHub address in the comments section of Moodle. I will grade your work by cloning your code and running it locally. I'll show you how this is done in class.