asgn01 Classes

Objectives

- Create a class
- Create class variables
- Create class methods
- Read class diagrams
- Use version control

Introduction

You will learn OOP (Object Oriented Programming).

For the first section of this class we will watch and code from the LinkedIn/Learning video

In this assignment we will

- Install Prettier for PHP (different than Prettier for HTML/JS)
- Use Git and GitHub
- Write the code challenge from chapter 1 in the video
- Write code for a similar program called bird-class

The Assignment

VS Code

Read this article on setting up Prettier for PHP File Formats in VSCode. Install the plugin according to the instructions. Note, you will need to change the `settings.json` file so you can run Prettier for multiple programs.

Folder structure

Create the following folder structure in your web root (htdocs).

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

Create a git repository called `web250`

For each of the exercises, use git for version control.

You may use the terminal or the GUI of your choice.

- Download and install git if necessary.
- Open your terminal
- Navigate to your root folder for this assignment

[Win]

```
cd c:/xampp/htdocs/web250
```

[Mac]

```
cd /Applications/MAMP/htdocs/web250
```

Initialize git

```
git init
```

Stage the files

The dot (.) is a wildcard and means stage **all** files

```
git add .
```

Commit the files

The "m" stands for "message"

```
git commit -m "Initial commit"
```

Watch the PHP videos

Watch chapters 1 and 2 from the PHP: Object-oriented Programming video and code along. I recommend watching the videos first, then go back and code with the author while watching a second time.

Complete the Challenge: Properties and methods. I realize that he provides an answer.

Commit changes

Commit changes. There is no hard-and-fast rule for this. It could be that you are at a breaking point, or you've written code that you don't want to lose.

Below is a common type of commit.

Substitute where you are at for X and Y

```
git add .
git commit -m"Completed X, next step is to Y"
```

Complete the Challenge: Properties and methods

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

Watch the Introduction and these chapters.

Introduction (download the exercise material).

Chapter 1.) Overview and Project Setup

Chapter 2.) 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 Challenge stage (add .) and commit your files.

Inside your `web250` folder, run the following commands. I've added `status` which is optional but I think you will find it helpful..

```
git add .
git status
git commit -m"Completed the bike challenge."
```

UML - Unified Modelling Language

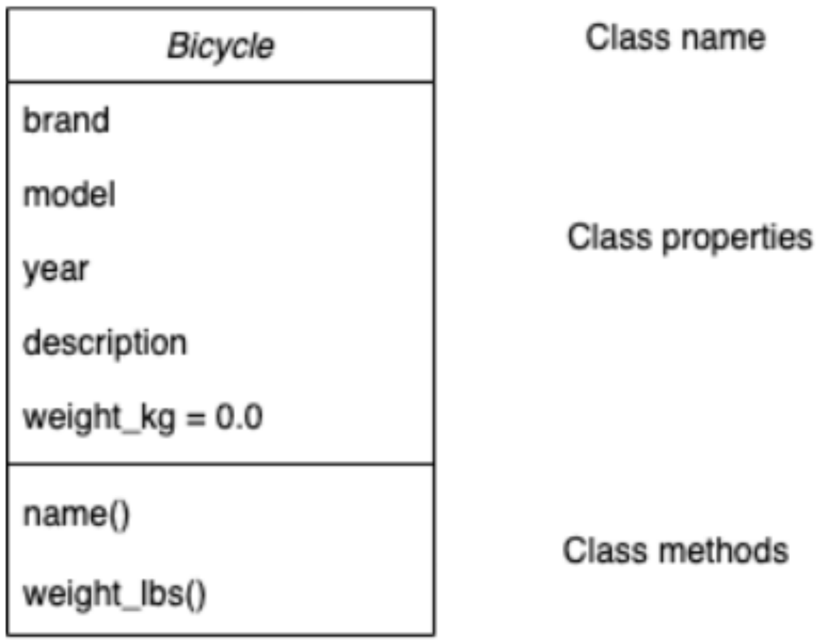
UML is a diagramming system comprised of many modeling techniques. We will use it for class diagrams just like the ones you saw in the OOD series. You can think of the class diagrams as a visual algorithm. These models are helpful in organizing your thoughts as your programs become more complex.

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 HTML code is dated. He is using HTML4 and also uses underscores in his file names.

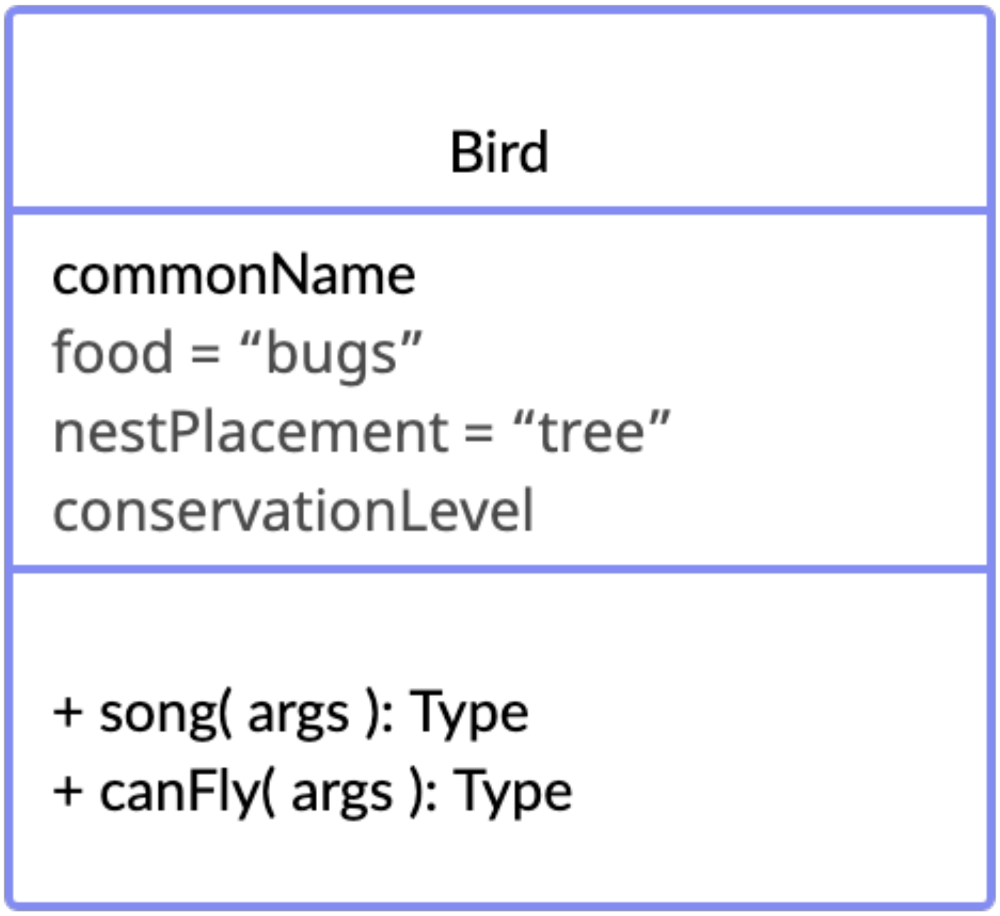
1. Change the HTMl them to fit the coding standards listed in Moodle.
2. Change the file names so they use hyphens instead of underscores.

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

2. Bird Challenge

- Create a new folder named `web250/bird-challenge`.
- Create a file named `index.php` Inside your `bird-challenge` folder.

Use the class diagram



(ignore the + signs for now). I have provided default values.

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. This is probably more frequent than you would usually commit, but it is good practice as you continue to learn version control.

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

$bird1

Note that the "code" below is from the class diagram. It is not using PHP syntax.

```
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
```

Display the content for both bird instances.

Git

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

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

GitHub

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

Create an account on GitHub if you don't have one. Create a new GitHub repo called **web250**.

Note: You may have to create SSH access. Follow the instructions in Git if necessary.

- Log on to GitHub
- Click on the New button
- Enter **web250** for the repository name
- Enter `This repo is for the school assignments at A-B Tech CC.`
- Click **Create Repository**
- Scroll to the section labeled, '…or push an existing repository from the command line.'
- Copy and paste the three lines displayed. Here is my example. Yours will be different.

```
git remote add origin https://github.com/charliekwallin/web250.git
git branch -M main
git push -u origin main
```

- Return to your GitHub account.
- Click on the link to your web250 repo at the top of the page to see the results.

Open your GitHub account to make sure everything pushed to your account. If it didn't, Git provides an 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.