Prepared by: n_ireland@fanshaweonline.ca

## HOMEWORK #2

# PHP CLASSES ASSIGNMENT

02_HOMEWORK

---

## Assignment Description:

You will have one week to create a multi-layer PHP class setup to describe some related data and functionality.

## Assignment Requirements:

Given the following related things, you'll need to create a hierarchy of PHP classes that represent the following objects:

- Animal
  - Canine
    - Dog
    - Wolf
  - Feline
    - Lion
    - Jaguar
    - HouseCat
  - Bird
    - Parrot
    - Cardinal

With the data above, you'll need three layers – the base class (**Animal**), the three sub-classes **(Canine, Feline**, **Bird),** and then the individual children classes (**Dog**, **Wolf**, **Lion**, **Jaguar**, **Tiger**, **HouseCat**, **Parrot**, **Cardinal**). You'll also need to ensure your classes are all namespaced and in folders similar to what we have done in class. Your parent (**Animal**) class must implement three methods: "**eat**", "**drink**" and "**speak**".

You'll also need to include at least 3 properties that describe what each animal likes to *eat*, what they *drink*, what *noise* they make when they speak. You'll need to overwrite / redefine these as necessary for each animal, ie. A **Dog** might eat "dog food" but a **Wolf** might eat "rabbits", etc.

Your **Feline** and **Canine** classes must also implement "**run**", and your **Bird** class must implement "**Fly**", Additionally, you'll need to overload (overwrite) methods as needed to make the functionality of each class specific to that animal; for example, with **Bird**, your **eat()** method could return something about eating seeds/bird feed, whereas your **Parrot** class' **eat()** might return something about eating fruit. You'll need to do this for all four required methods - "**eat**", "**drink**", "**speak**", and "**move**". Returning a string like we have been doing in class is sufficient, but you're free to get more complicated if you would like. You must have at

least one method where you're calling the parent's implementation and using that value in your method, similar to what we had done in lecture.

Additionally, you'll need to implement any traits you feel necessary (minimum one) to share functionality / methods between classes – for example, any functionality that might be similar between **Feline** and **Canine**, but doesn't apply to all animals. You'll also need to implement a **Park** class that accepts three different types of animals and prints some messages about them, similar to the **Meal** class we implemented in lecture.

Finally, you'll need a **public** folder and an **index.php** that can be used to test your code, with the **spl_autoload_register** implementation we used in lecture. You'll also need to **var_dump** out each of the classes you've created so they can be inspected, similar to what we did in lecture.

All code you write must be commented describing the functionality and explaining what is happening. You must also try and make your variable names, function names, etc. as descriptive as possible, based on what value they are holding, what function they perfom, etc.

You are permitted to use online resources – tutorials, references, etc., however copying *directly* from said resources (including my lectures, within reason – ie. don't directly copy what we've done in class and submit it, modify it and make it your own) will result in a failing grade – make sure you understand the code you're referencing, and make it your own. You're required to list any additional references / resources used in the README.md file in your github repository (see **Submission** for more information).

Prepared by: n_ireland@fanshaweonline.ca

## Assignment Rubric:

**0%** - I can't clone the github repository (ie. You haven't given me access correctly – see **Submission** for more information), or your references / **README** are inaccessible.

**0% - 30%** - You have a seperate file for each class, and you've followed a similar folder structure to what we have in class; starting with **includes/classes** and including your files in there. You've implemented all the required methods (as listed above) but there is no inheritance and no hierarchy. You're manually importing each class individually using **require_once**.

**30% - 60%** - Your classes extend one another to create the hierarchy. You're overwriting methods as needed to adjust functionality, and implementing methods as few times as needed. You've implemented your **Park** class, and it prints some messages about the three animals it is created with. You're using **spl_autoload_register** to automatically import your classes.

**60% - 90%** - All of your individual classes are namespaced appropriately, and are still being automatically imported. You've moved classes into namespaces / folders that group them together based on their hierarchy. Your code is documented appropriately with doc blocks for all your methods and parameters.

**90% - 100% -** You've included one or more traits to re-use methods / functionality across multiple classes. Your code is well documented and has the appropriate typehints for method parameters and return types. Your **Park** class uses typehints to prevent incorrect or non-unique animals from being passed in.

## Submission:

Your code will need to be submitted within a *private* github repository. Please note that if *your repository is not private, or you are unable to add me correctly, your assignment will receive a mark of* ***0 / 100***.

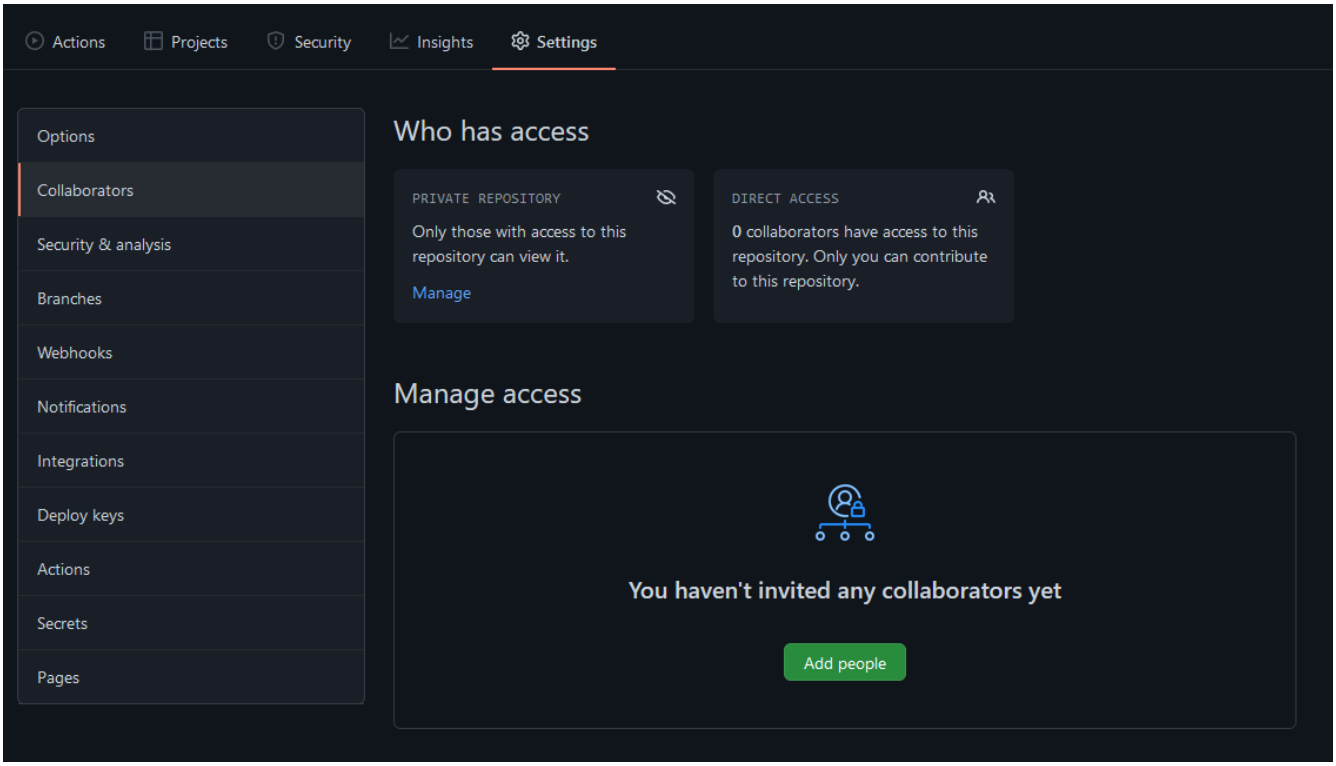
Please name your repository with something similar to the following format:
**full_name_MMED-3014-homework_2**
For example: **nicholas_ireland-MMED-3014-homework_2**

You'll need to include a **README.md** file in your repository that lists all the references you used for your project. Please list them in the following format:

**Reference Link:** https://myreference.com
**Reference Used:** my/path/to/a/file.php lines 10-20, lines 50-60

Please submit the link to your github repository in the "Text Submission" box on FOL. You'll then need to invite me to your repository as a contributor. Here's some quick instructions in case you're unfamiliar:

Head to your repository settings ("**Settings**") on the far right of the top bar.
From here, you'll need to select "**Collaborators**" in the left sidebar:




Then, click the "**Add people**" button at the bottom of the page.
Then, enter my email – **n_ireland@fanshaweonline.ca** – and hit click the "Add" button:

That's it! From there, I should have access to your repository. Please copy the URL to your repo and paste it into the text field in the submission so I am able to find it.

## Additional Information:

Missed tests/exams will not be rescheduled without some valid evidence of some important event over which the student has no control (e.g., Court appearance, death in the family). Missed tests or exams, therefore, can receive a zero. The students are advised to notify the professor prior to missing the test.

Students are expected to hand in all assignments to the course instructor on the due date, and all assignments must be submitted in the format specified by the instructor (e.g., on FOL, in printed form, on a specific lab computer, etc.); assignments will not be accepted in any format other than that specified.

Late assignments will not be accepted, nor will make up test or assignments be permitted, without some valid evidence of some important event over which the student has no control (e.g., documented illness, death in the family). Missed tests or assignments, therefore, will receive a mark of zero. Late assignments and make-up tests will only be permitted following the submission of adequate documentation acceptable to the instructor (e.g., a doctor's note). Students are advised to notify the instructor prior to missing an assignment due date or a scheduled test.

Immediately upon return from an illness/absence in which a test or assignment has been missed, the student is responsible for contacting the course instructor to discuss the problem. The instructor will make arrangements for any student deemed eligible. The alternative test/assignment will be of equal value to the one missed with no grade penalty. The timeline and due dates will be determined by the course instructor.

At mid-term, any unsatisfactory results will be reported to the student.

This course may be revised by the professor with suitable notification to the students. Students are responsible for making arrangements to pick up missed handouts, assignments and course announcements from classmates.

Plagiarism (e.g., failure to acknowledge sources used, submitting another student's work under your name, or producing work for another student to submit) is a serious academic offense that shall result in appropriate penalties, to be determined at the discretion of the course professor in consultation with the chairperson of the Communication Arts division. The penalties shall range from failure of an assignment to possible failure of the course. Students shall not make the assumption that any provision will be made by the professor to permit the student to rewrite or redo failed assignments.