

asgn02 -- Inheritance and object access control

Objectives

- Use inheritance in your code
- Use object access control (public, protected, private) in your code

Watch the tutorials

Watch the next two chapters in LinkedIn/Learning tutorial, [PHP: Object-oriented Programming](#)

1. Class Inheritance
2. Object Access Control

Set up

- Create a folders named **asgn02-inheritance** inside your **web250** folder.
- Create files named **challenge_03.php**, **inheritance.php**, and **oac.php** inside your **asgn02-inheritance** folder.

Site note: we still are not to the point of creating files just for **classes** but that is coming soon.

Git

Class Discussion (not graded)

Class discussion on using one Git repo for the entire WEB 250 class or using a repo for each of the assignments. We can extend this in Slack.

Once your files are set up, it is time to start version control.

In the terminal, navigate to your `web250/asgn02` folder and use the following git commands

```
git status
git add .
git commit -m"Starting asgn02-inheritance"
```

Create a branch

In the past assignment we worked on the `master` branch. The `master` branch is typically use for code that is ready for production. Once you get more comfortable with git, you will want to create branches for bug fixes, features, etc.

Read the docs on [Git branches from Atlassian](#)

First, see what branches are available. Hint (it is only `master`).

```
git branch
```

Type `q` to exit the screen.

Create a branch called `dev` then checkout to it.

```
git branch dev
```

This only creates the branch. If you type `git branch` again you will see it is there, but you will still be on the `master` branch. You need to checkout the branch.

```
git checkout dev
```

Now you can work safely on this branch without possibly messing up your code on your `master` branch.

1. Class Inheritance

Watch all of the **Class Inheritance** videos. It is up to you if you would like to code along. I find that it helps. I want to emphasize, that I get the most out of the videos (or reading text) if I do all of it first, then go back in a second time and start working with the code.

The Challenge

Complete the challenge at the end. Choose your subject as he has suggested. I have found that anything with a solid taxonomy is a good candidate for inheritance. Mr. Skoglund mentions

animals, in addition other similar topics such as birds or plants would make for good topics. There are great resources for birds at [Cornell Lab of Ornithology](#) and the [National Audoban Society](#). Be careful that you don't make this too big -- in can grow out of control quickly.

This is hard. Make sure you give it some time.

Your Challenge Code Must

- Have one parent class and at least two subclasses.
- Classes must contain at least one class variable and one class method.
- Demonstrate that the subclass inherits from the parent class.

Git Merge

Once your code is complete, it is time to merge it with the `master` branch. There are a couple of ways to do this. I do it by checking out to the `master` branch, then I perform the merge. First you will need to stage and commit.

```
git status (I just do this to see where I am at)
git add .
git commit -m "Completed the inheritance challenge"
git status (same reason as before)
git checkout master
git merge dev
```

Once that is complete you will want to checkout to the `dev` branch to continue working

```
git checkout dev
```

2. Object Access Control Challenge

Watch the entire Object Access Control tutorial.

Use the **challenge_03.php** file for this code.

You must address all of his points.

1. Add visibility modifiers to the bicycle class
2. Set visibility for all existing properties and methods. Deciding what to make public, protected and private is difficult. To give yourself some help, check out his answer for the

visibility he has set, then try to do it yourself.

3. Create a unicycle subclass
4. Add the property `$wheels` and set values for each of the classes.
5. Define a `wheel_details` method which returns "it has 2 wheel" when called on an instance.
6. Make `weight_kg` a private property.
7. Define a `set_weight_kg()` method (setter method).
8. Create a getter method to read that value back followed by "kg".
9. Modify the `weight_lbs()` method to add "lbs" to it.
10. What bug have you introduced to the `$weight_kg` ?

Try to do as much of this as you can without peeking at the solution. It's a great exercise.

Git

Once it is finished, then it is the same process with git

```
git status (I just do this to see where I am at)
git add .
git commit -m "Completed the challenge03"
git status (same reason as before)
git checkout master
git merge dev
```

Checkout to the `dev` branch to continue working

```
git checkout dev
```

3. Add Object Access Controls to Your Code

Modify the code you wrote for the Inheritance exercise. Use the same principles the Mr. Skoglund used to improve it by adding visibility modifiers (public, protected, private).

Try using the private modifier for a class variable in the superclass and see what happens in the subclass.

Try using the protected modifier for a class variable in the superclass and see if that corrects the problem.

4. Add setters and getters to your code

At first this will seem like overkill for such a small program (it is), but it is a good starting point. Remember one of the key concepts is to set class variables as private and the setter and getter functions as public.

Git

Same process as before

GitHub

Your code is now ready for production, let's push it to the GitHub repo.

Create an account at [GitHub](https://github.com) if you haven't already.

NOTE: Git has moved away from using the term `master` and now used `main` for the main branch.

Create a repo

- Click on the green **New** button.
- Name your repo **asgn02-inheritance**
- Add **OOP school work for class at A-B Tech CC** for the description.
- Click **Create repository**

You have already completed a lot of the steps in the section titled **...or create a new repository on the command line**, so scroll down to **...or push an existing repository from the command line**

Here are the lines from GitHub that you will need.

NOTE: Do not copy mine verbatim as it contains my username.

```
git remote add origin https://github.com/charliekwallin/asgn02-inheritance.git
git branch -M main
git push -u origin main
```

Submit

Copy the URL for your GitHub account and post it in the Comments section for the Moodle

assignment.

I will pull your code from GitHub to run it and read the code.

Coming up

Domain name and Web host

If you are comfortable with the following concepts, please go ahead and try to set them up. If not, we can wait until next week.

- Push your code to your host
- Make sure you have an SSL certificate so you can
 - Use SFTP
 - Create HTTPS protocol