# Homework Assignment

This assignment helps us assess your programming abilities. Your mission, should you choose to accept it, is to complete the tasks below and return your completed files to us within five business days. Keep in mind: this is our **sole visibility** into your ability to understand requirements and implement technical solutions. Although you’re working with very simple objects, use best practices and avoid taking shortcuts. We take this seriously.

## Setup Tasks

1. Work in PHP on the operating system of your choice. Feel free to use your favorite IDE.
2. Translate the three .pseudo files, as-is, into PHP. Retain and submit these initial translations along with your final work.
3. Run your code and verify the results against this output:

Name is currently

Name has been changed to Garfield

Inserting Garfield into table Cat

## Coding Tasks

### Part 1

As you proceed through these tasks, modify the main.php program as you see fit, just to informally demonstrate the additional functionality. (Formal tests come later.)

1. Modify the constructor to always initialize the cat’s age to a random number between 5 and 10.
2. Modify the constructor to optionally support an initial name for the new cat.
3. Add a method called **speak**() to make the cat say “meow”.  
   Note: print the word to STDOUT.
4. Modify the **speak**() method to accept an optional argument.  
   If not supplied, the cat still says “meow”.  
   If supplied, the cat prints the value of the argument.
5. Modify the **setName**() method to keep a list of all the names the cat has ever had.
6. Create a **getNames**() method to return a list of all the names the cat has ever had.
7. Modify the class so that the cat’s age increases by 1 every five times it speaks.
8. Create a new method **getAverageNameLength**() that returns the average length of all the names the cat has ever had.
9. Copy your Cat code to make a Dog object, making necessary dog-like changes along the way.

### Part 2

Now it’s time to actually do something with your objects. Create a new script, petShop.php, which implements the following three functions. (**Important**: do not create a physical database to store your objects; only make sure Data.php pretends to do so. The “db” attribute represents a database access layer, such as PDO.)

1. Create a function called **saveTest**() to:
   1. Create a cat with a name and insert it into the database.
   2. Create a dog with a name and insert it into the database.
2. Create a function called **savePetShop**() to:
   1. Create three nameless cats.
   2. Create three nameless dogs.
   3. Insert all six pets into the database.
   4. Important: guarantee that all six pets (or no pets) are persisted.
3. Create a function called **logStats**() to:
   1. Print interesting information about your script’s execution results to STDOUT.

## SQL Tasks

In a new file:

1. Write SQL statements to create table(s) to store our animals.  
   Hint: Don’t worry about the actual DBMS or SQL dialect. Generic SQL is fine.  
   Hint: Do ensure your database schema could be used to fully persist an instantiated object.
2. Provide sample insert statements.

## Testing Tasks

In a new file (or files):

1. Create unit tests to assert that a cat’s initial age is always between 5 and 10.
2. Create unit tests to assert that cats can indeed be given a name at birth.
3. Create unit tests to assert the **speak**() method is properly functioning.
4. Create any other unit tests you deem appropriate.

## Documentation

As a final step, create a brief README file for the project. Feel free to use this file to:

1. Describe the project, file structure, what it does, etc.
2. Discuss how or, more importantly, why you did things the way you did.

In short, anything you think would help a future developer understand your project.