# Challenge 2: Implement an Animal Class

In this lesson, you will implement an animal class and its subclasses from scratch.

#### WE'LL COVER THE FOLLOWING ^

- Problem Statement
  - Input
  - Sample Input
  - Sample Output

## Problem Statement #

The code below has:

- A parent class named Animal.
  - Inside it *define*:
    - name
    - sound
    - \_\_init\_\_()
    - Animal\_details() function:
      - It prints the name and sound of the Animal.
- Then there are **two derived** classes
  - o Dog class
    - has a *property* **family**
    - has an initializer that calls the parent class initializer in it through super()
    - has an overridden method named Animal\_details() which prints detail of the dog.
  - Sheep class

- has a property color
- has an initializer that calls the parent class initializer in it through super()
- has an overridden method named Animal\_details() which prints detail of the sheep.
- The **derived classes** should override the Animal\_details() method defined in the Animal class.
  - The overridden method in Dog class should print the value of family as well as the name and sound.
  - The overridden method in Sheep class should print the value of
     color as well as the name and sound

#### Input #

- name of Dog is set to **Pongo**, sound is set to **Woof Woof** and family is set to **Carnivore** in the initializer of Dog object.
- name of Sheep is set to **Billy**, sound is set to **Baaa Baaa** and color is set to **White** in the initializer of Sheep object.
- Now, call Animal\_Details() from their respective objects.

Here's a sample result which you should get.

#### Sample Input #

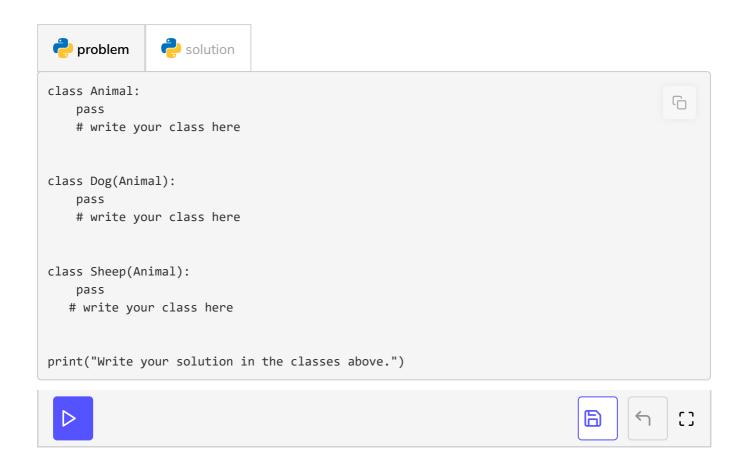
```
d = Dog("Pongo", "Woof Woof", "Husky")
d.Animal_details()
print(" ")
s = Sheep("Billy", "Baaa Baaa", "White")
s.Animal_details()
```

### Sample Output #

```
Name: Pongo
Sound: Woof Woof
Family: Husky

Name: Billy
Sound: Baa Baa
```

#### Color: White



Solution will be reviewed in the next lesson.