

OOP and Inheritance

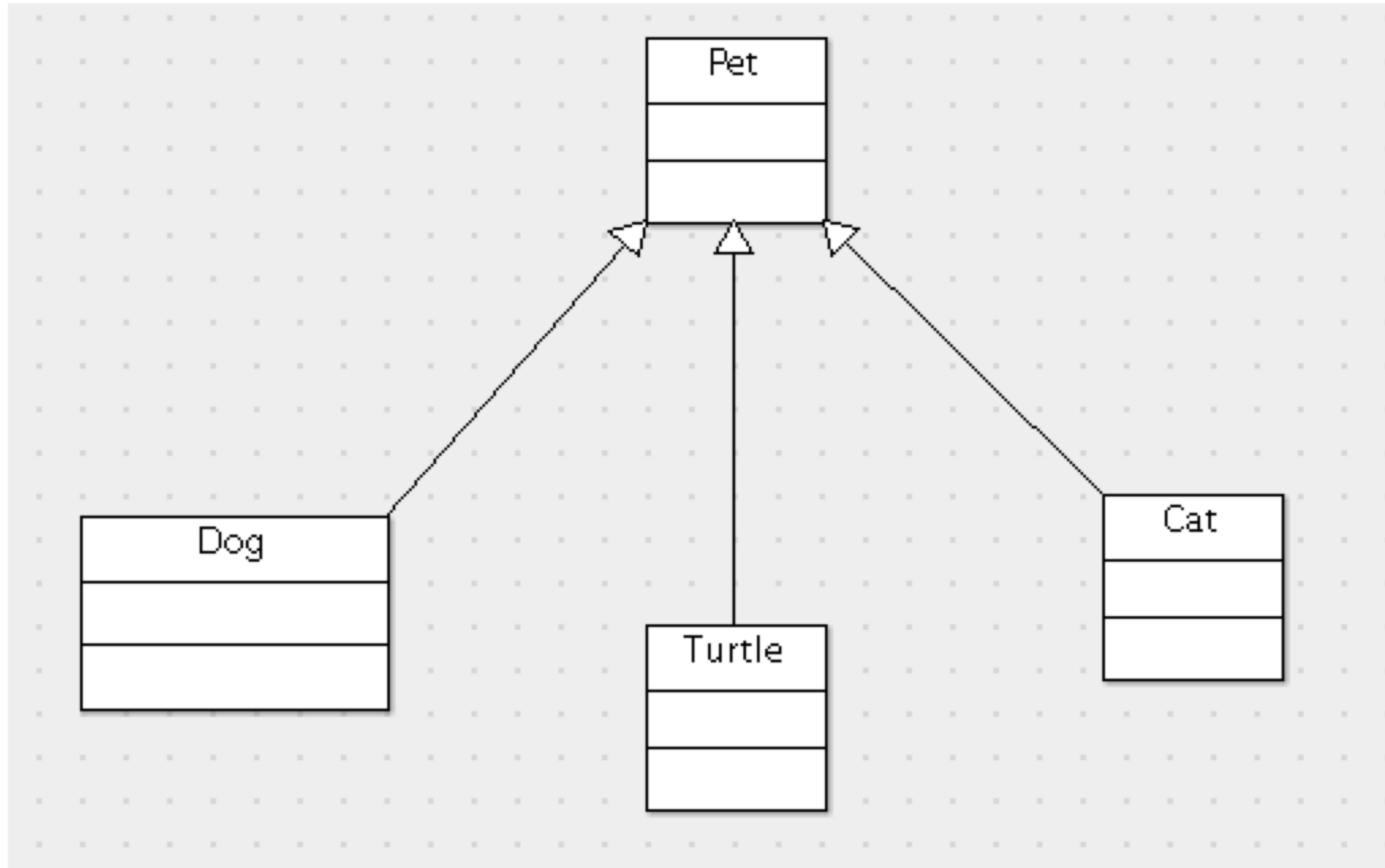
Pet Store

```
class Dog:
    def __init__(self, name):
        self.name = name
    def describe(self):
        return self.name + " says: Woof!"
class Bird:
    def __init__(self, name):
        self.name = name
    def describe(self):
        return self.name + " says: Chirp!"
class Cat:
    def __init__(self, name):
        self.name = name
    def describe(self):
        return self.name + " says: Meow!"
```

```
>>> coco = Bird("Coco")
>>> coco.describe()
'Coco says: Chirp!'
```

Repetitive! How can we do better?

Common Base



Inheritance

Dog, Cat, and Bird are all specific versions of a more general concept: a **pet**.

When you have several classes related in this way, you can create a **base class** called Pet.

This base class holds common code that Cat, Dog, Bird, etc. all build on, and use.

This is called **inheritance**.

Inheriting

Use parentheses on the `class` line.

```
class Pet:
    def __init__(self, name):
        self.name = name
class Dog(Pet):
    def describe(self):
        return self.name + " says: Woof!"
class Cat(Pet):
    def describe(self):
        return self.name + " says: Meow!"
```

```
>>> fido = Dog("Fido")
>>> fido.describe()
'Fido says: Woof!'
>>> fluffy = Cat("Fluffy")
>>> fluffy.describe()
'Fluffy says: Meow!'
```

Terminology

We say that Dog and Cat **subclass** Pet. And, Dog and Cat are **subclasses** of Pet.

Conversely, Pet is the **superclass** of both Dog and Cat.

Pet is also the **base class** because it doesn't inherit from anything.

The describe() method

How can we put describe() in the Pet class?

We can do this by using a **class-defined variable**.

```
class Pet:
    # We can put a default "sound" variable here.
    sound = ""
    def __init__(self, name):
        self.name = name
    def describe(self):
        return self.name + " says: " + self.sound

class Dog(Pet):
    # The subclass can override the sound variable.
    sound = "Woof!"

class Cat(Pet):
    sound = "Meow!"
```

```
>>> fido = Dog("Fido")
>>> fido.describe()
'Fido says: Woof!'
>>> fluffy = Cat("Fluffy")
```

Lab: OOP

Lab file: `oop.py`

- In `labs` folder
- When you are done, study the solution - compare to what you wrote.

Instructions: `LABS.txt` in courseware.