Computing

Lesson 4: Inheritance

KS4 Object-Oriented Programming

Mac Bowley

[1] *Materials from the Teach Computing Curriculum created by the National Centre for Computing Education*

Identifying object-oriented code

```
1 from vehicle import Vehicle
2
3 car = Vehicle(“BMW”, 4, 120)
4
5 print(“My %s has a top speed of %i mph“ % (car.getName(), car.getSpeed()))
6
7 car.drive()
```

Have a look at the code above, can you give an example of the following?

- A class
- An object
- An attribute
- A method

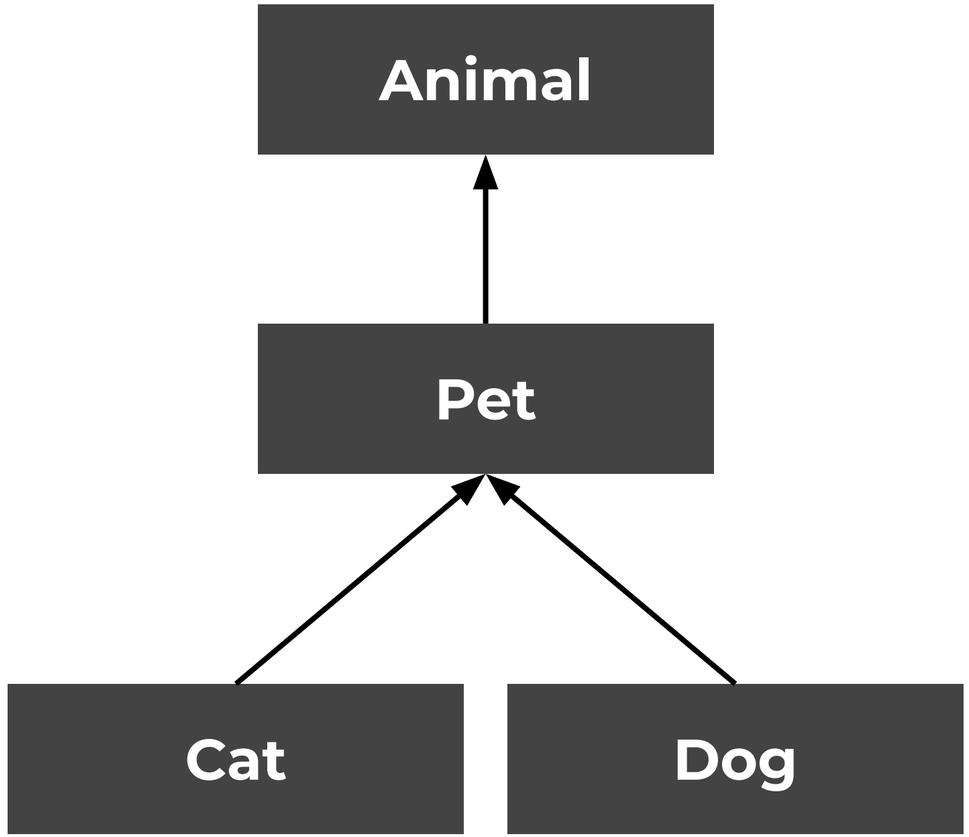
Bonus challenge - What is the function on **line 3** called?

Task title

Which class would this **attribute** belong to?

Does it apply to all or just some of them?

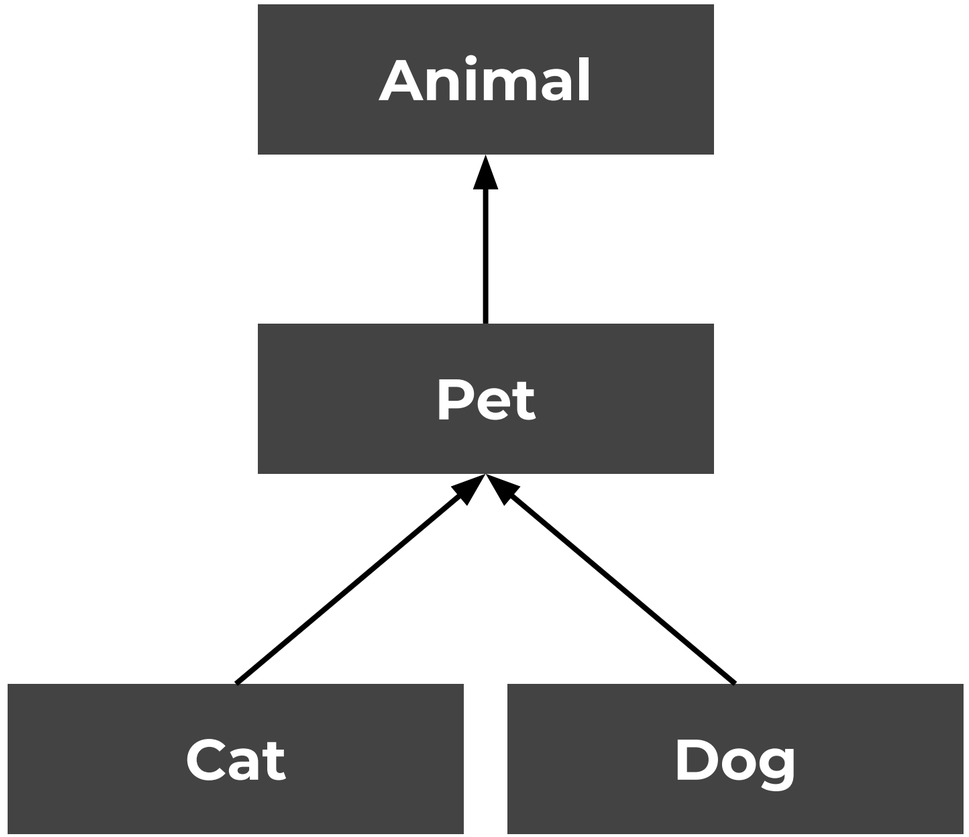
Number of legs



Task title

Which class would this **method** belong to?

Does it apply to all or just some of them?



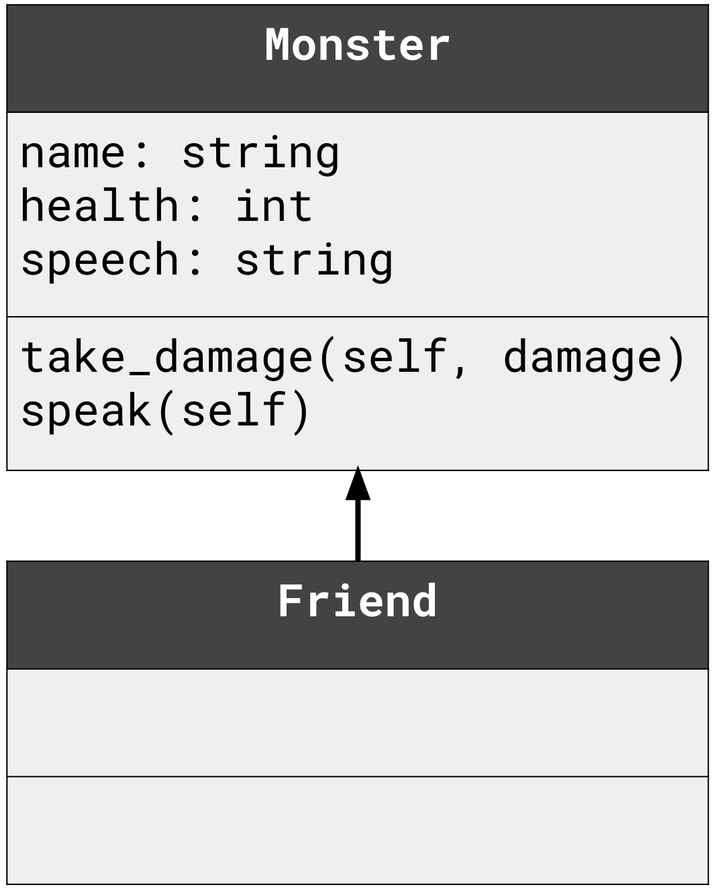
Wag tail

Improving Monster Quest

The Friend subclass needs to...

1. Say something when the player high-fives them
2. Become friends with the player and give a gift.

What attributes does this new subclass need?

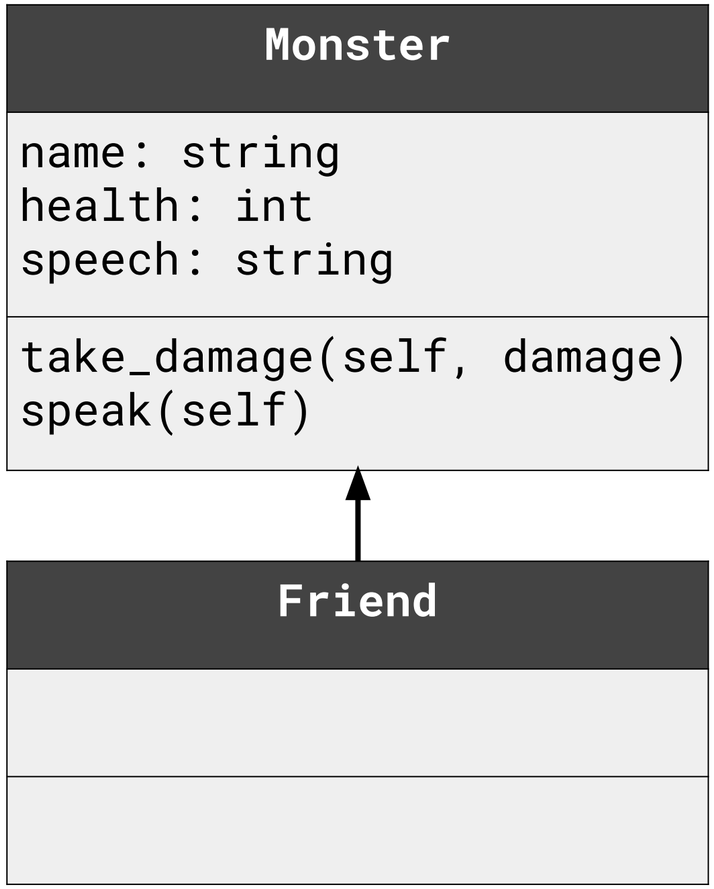


Improving Monster Quest

The Friend subclass needs to...

1. Say something when the player high-fives them
2. Become friends with the player and give a gift.

What methods does this new subclass need?



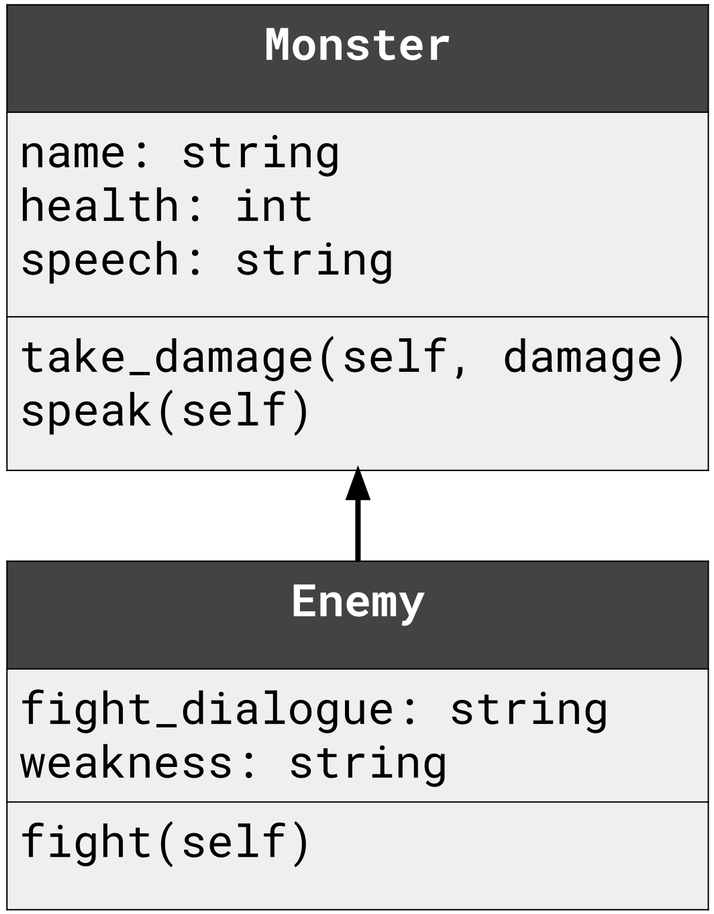
Creating the Friend class

```
 1 class Friend(Monster):
 2
 3   def __init__(self, name, health, speech, highfive_dialogue, gift):
 4     super().__init__(name, health, speech)
 5     self.highfive_dialogue = highfive_dialogue
 6     self.gift = gift
 7
..   GETTERS AND SETTERS
19
20   def highfive(self):
21       print("%s raises their hand and says…" % (self.name))
22       print(self.highfive_dialogue)
23       print("%s gives you %s" % (self.name, self.gift))
24
```


Improving Monster Quest

The Enemy class needs to:

- Have a weakness
- Say something when the player fights them
- Ask the player to choose a weapon of choice in the fight
- If the player's weapon is the same as the weakness then the player wins
- Otherwise the player loses.



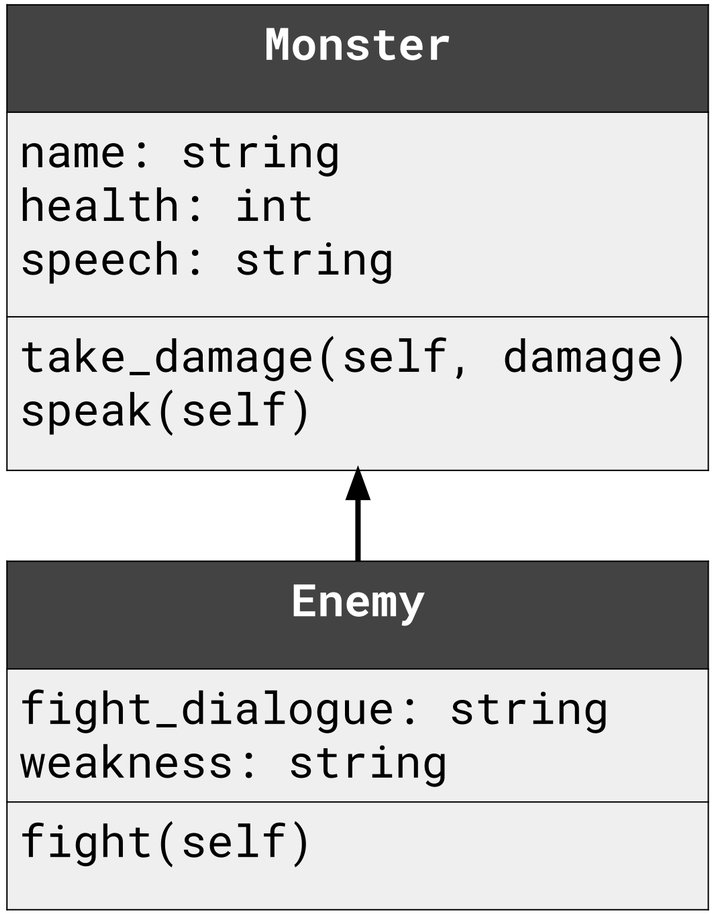
Improving Monster Quest

```
1 class Monster():
```

This is how you **declared** the Monster class.

What do you need to change for the new Enemy class?

Add the class definition and the constructor to your plan now.

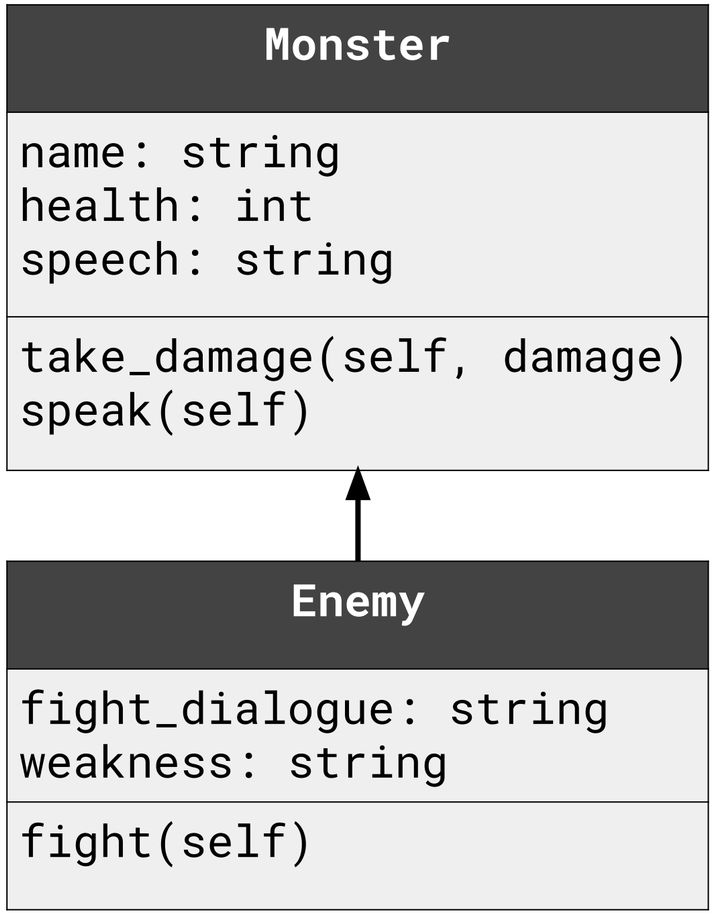


Improving Monster Quest

Next you need to add getters and setters, but not for all the attributes.

Which getters and setters will the Enemy class inherit?

Add the other getters and setters to your plan now.

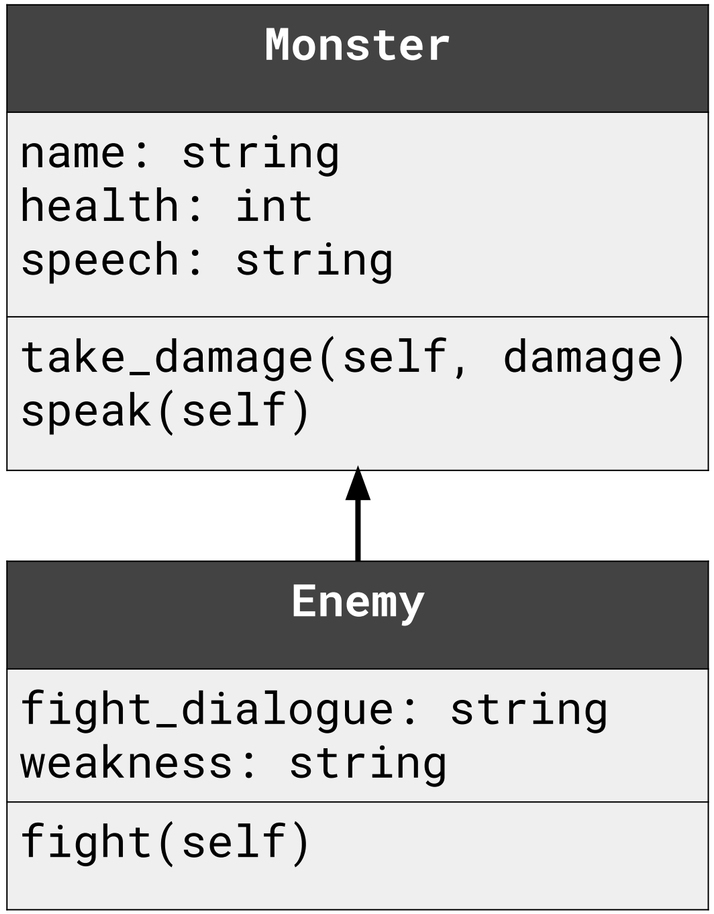


Improving Monster Quest

There is one new method you need to add to the Enemy subclass - the fight() method.

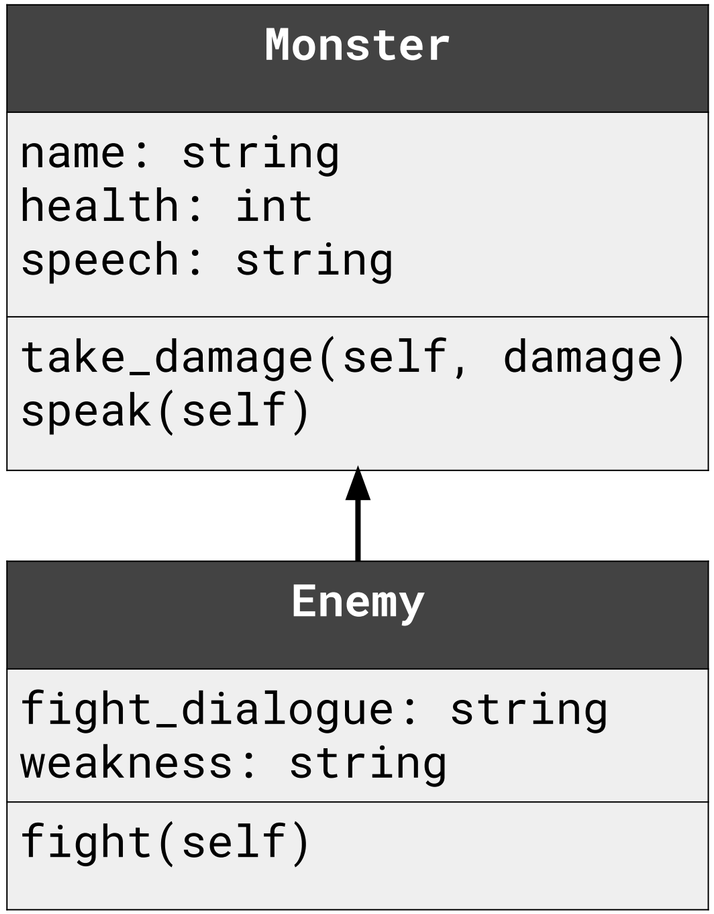
What parameters do you need for this method?

Use this and the requirements from before to plan the new method now.



Improving Monster Quest

Use your plan to write the new subclass into your Python repl.



Identifying object-oriented code

```
1 from monster import Monster, Friend, Enemy
2
3 zombie = Friend("Dave", 100, "Braiiinnnnssss!", "Arrrghhh!!", "Cheese")
4
5 vampire = Enemy("Gustav", 150, "I vant to suck your blood", "Hissss!!", "Garlic")
```

Now it is time to test your new creations!!

Add lines like these to your **main.py** file.

Then try calling the new methods, but also try using the Monster methods **speak** and **take_damage** too!