Introduction to Object-Oriented Programming

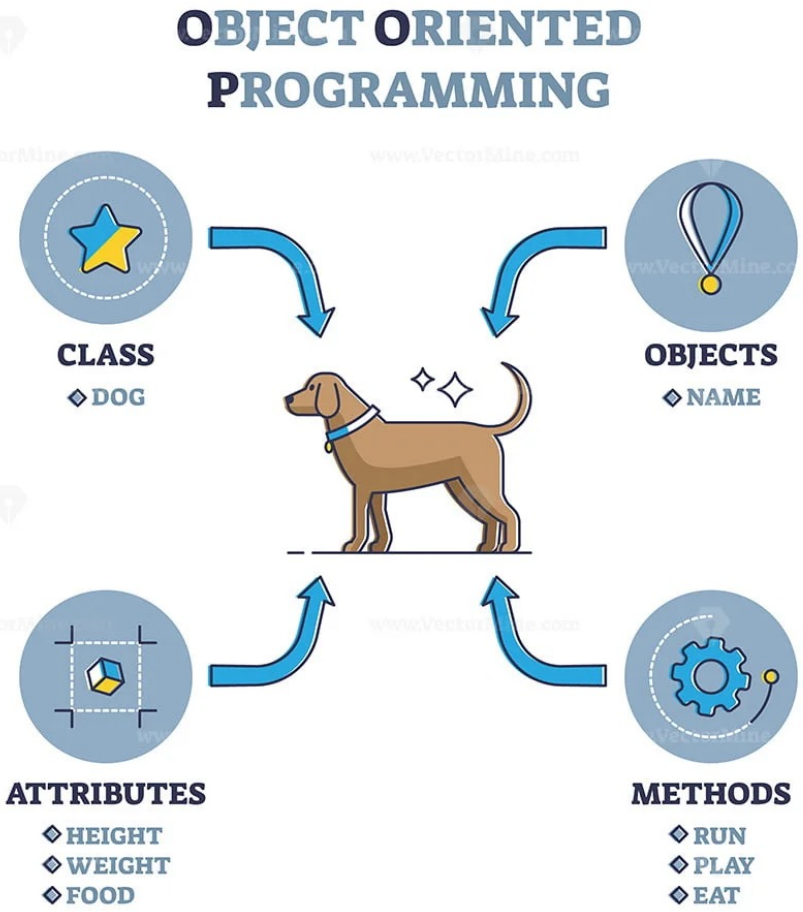
Week 1

What is Object-Oriented Programming?

Object-oriented programming (OOP) is a way of writing programs that organizes the code into reusable pieces called "objects"

These objects represent real-world things.

They have both data (attributes) and actions (methods) associated with them.



https://www.online-python.com

Features of OOP

Classes & Objects

Classes

- Blueprint for creating objects, providing initial value (variables) and implementations of behaviour (methods)

Objects

- Instance of a class.
- Do not have to know all the internal details of the code.

```
1  class Rabbit:
2      pass
3
4  # Creating an instance of the Rabbit class
5  rabbit_instance = Rabbit()
```

Variables/Attributes

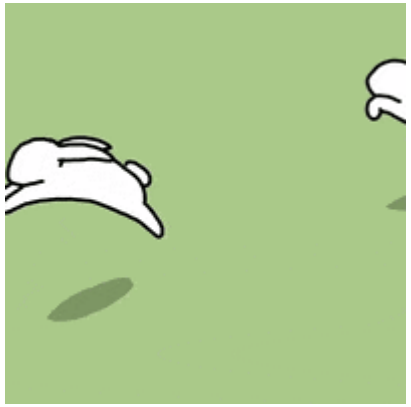
```python
class Rabbit:
    # Class attributes
    tail = "fluffy"
    ears = "long"

    def __init__(self, name):
        # Instance attributes
        self.name = name

# Accessing class attribute
print(Rabbit.tail)  # Output: fluffy
print(Rabbit.ears)  # Output: long

# Creating instances and accessing instance attributes
bugs = Rabbit("Bugs")
print(bugs.name)  # Output: Bugs

thumper = Rabbit("Thumper")
print(thumper.name)  # Output: Thumper
```

- Characteristics of an object
- Passed in as arguments to an object
- For example, a Rabbit object has several pieces of information associated with it that a user can extract
  - A class attribute, which is the same for every rabbit, i.e. fluffy tail and long ears
  - An instance variable, i.e. name, which is unique to every rabbit

Methods/Procedures

- Equivalent to a function.
- E.g., the Rabbit object might have a method for running (hop).

```
1   class Rabbit:
2       # Class method
3       def run(self):
4           print("hop")
5
6   # Creating an instance of the Rabbit class
7   rabbit_instance = Rabbit()
8
9   # Accessing class method using the instance
10  rabbit_instance.run()
```



```python
class Superhero:
    def __init__(self, name, strength, power, costume, secret_identity, points, health):
        self.name = name
        self.strength = strength
        self.power = power
        self.costume = costume
        self.secret_identity = secret_identity
        self.points = points
        self.health = health

    def is_attacked(self):
        #print self.health #uncomment print statements to see how variables are updated
        self.health -= 1
        return self.health

    def heal(self):
        #print self.health
        self.health += 2
        return self.health

    def gain_points(self):
        self.points += 1
        return self.points

batman = Superhero("Batman", 3, "fight", "black", "Bruce Wayne", 3, 5)
```

Exercise

Task 1: Define the Dog Class

Create a Python class named Dog with the following specifications:

- Add a class attribute named attr1 and set its value to the string "mammal".
- Define an instance attribute name which will be initialised through the constructor.
- Implement a method named speak() which prints "My name is [dog's name]" where [dog's name] is replaced with the value of the name attribute.

Task 2: Create Dog Instances

Instantiate two objects of the Dog class with the following names:

- Rodger
- Tommy

Task 3: Access Class Methods

Call the speak() method on each object created in Task 2.

Challenge: Create a new class

Create a new class (of your choosing)! And add in class attributes, instance attributes and methods that describe the class

Solution

```
class Dog:

    # class attribute
    attr1 = "mammal"

    # Instance attribute
    def __init__(self, name):
        self.name = name

    def speak(self):
        print("My name is {}".format(self.name))

# Driver code
# Object instantiation
Rodger = Dog("Rodger")
Tommy = Dog("Tommy")

# Accessing class methods
Rodger.speak()
Tommy.speak()
```

Sources

- GeeksforGeeks (2016). *Python OOPs Concepts*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/python-oops-concepts/.
- brilliant.org. (n.d.). *Object-Oriented Programming | Brilliant Math & Science Wiki*. [online] Available at: https://brilliant.org/wiki/object-oriented-programming/.
- ocoyucanpuebla.gob.mx. (n.d.). *Ayuntamiento de Ocoyucan*. [online] Available at: https://ocoyucanpuebla.gob.mx/?q=object-oriented-programming-or-oop-paradigm-explanation-outline-diagram-zz-EMjmo0Hz [Accessed 5 Apr. 2024].
- GIPHY. (n.d.). *Rabbit Hopping GIFs - Find & Share on GIPHY*. [online] Available at: https://giphy.com/explore/rabbit-hopping [Accessed 5 Apr. 2024].
- Press, A. (n.d.). *Today in History, July 27, 1940: Bugs Bunny first appeared in Warner Bros. cartoon*. [online] The Enquirer. Available at: https://eu.cincinnati.com/story/aerial-journalism/2019/07/27/today-history-july-27-1953-armistice-ends-korean-war/1827764001/ [Accessed 5 Apr. 2024].
- Disneyclips. (n.d.). *Thumper Clip Art*. [online] Available at: https://www.disneyclips.com/images5/bambi4.html [Accessed 5 Apr. 2024].
- Batman Wiki. (n.d.). *Batman (Lego Movie)*. [online] Available at: https://batman.fandom.com/wiki/Batman_(Lego_Movie) [Accessed 5 Apr. 2024].
- Sinker, M. (n.d.). *Beatrix Potter*. [online] 4columns.org. Available at: https://4columns.org/sinker-mark/beatrix-potter [Accessed 5 Apr. 2024].