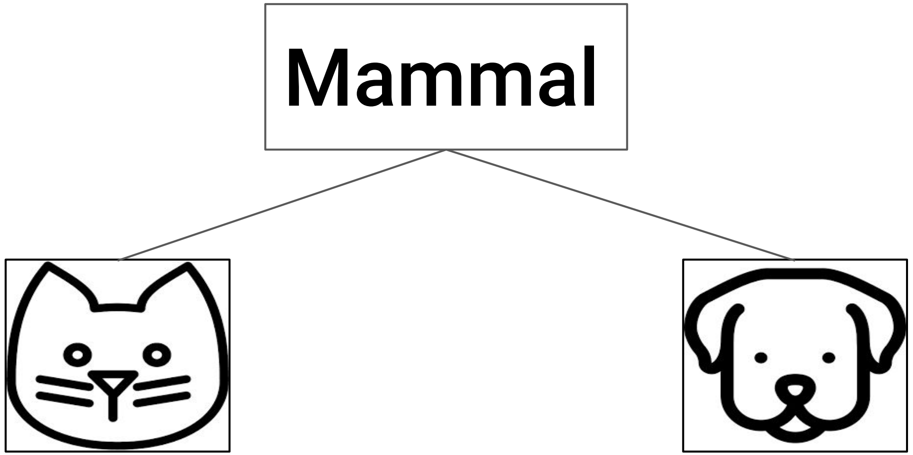
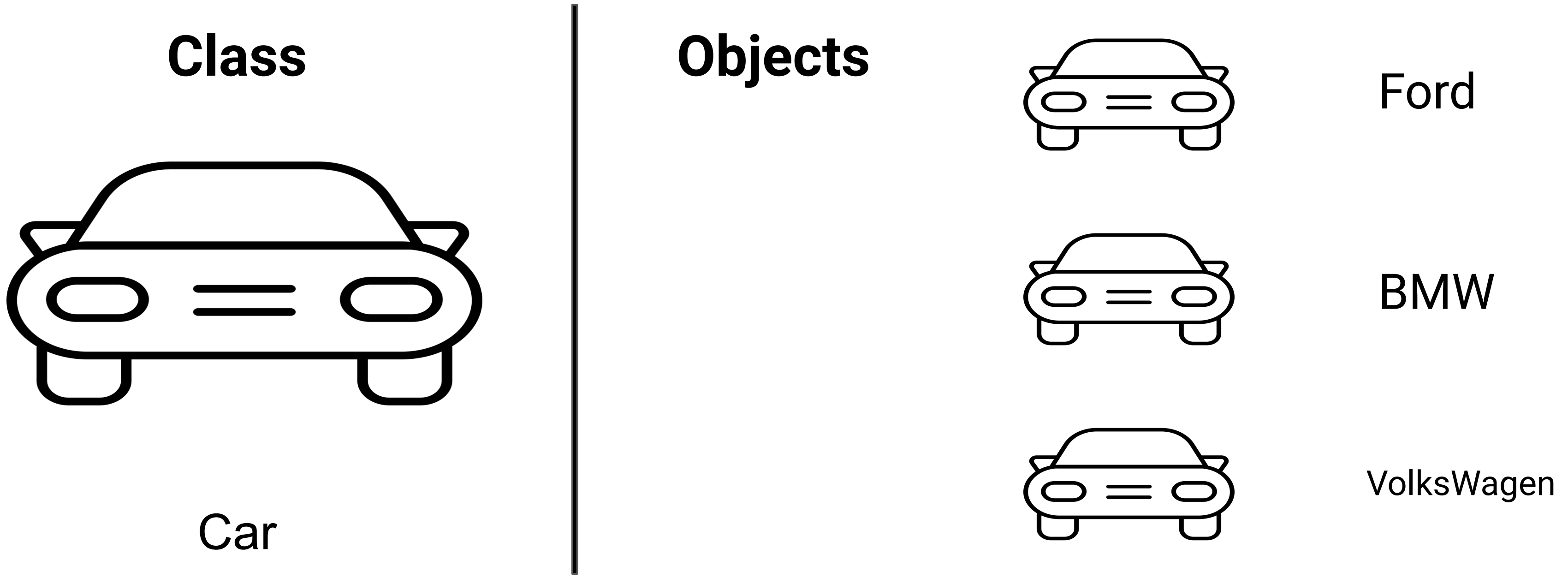
Object-Oriented Programming



Izundu Dan-Ekeh

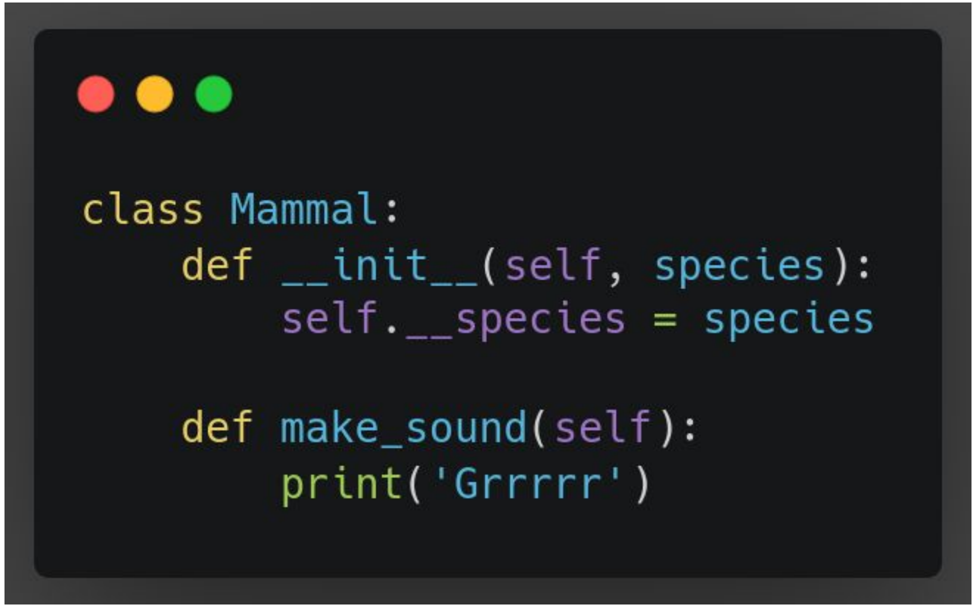
Object-Oriented Programming (OOP) is a programming paradigm that is centered on creating ***classes*** and ***objects***.



Class

A class is a blueprint or template for creating objects.

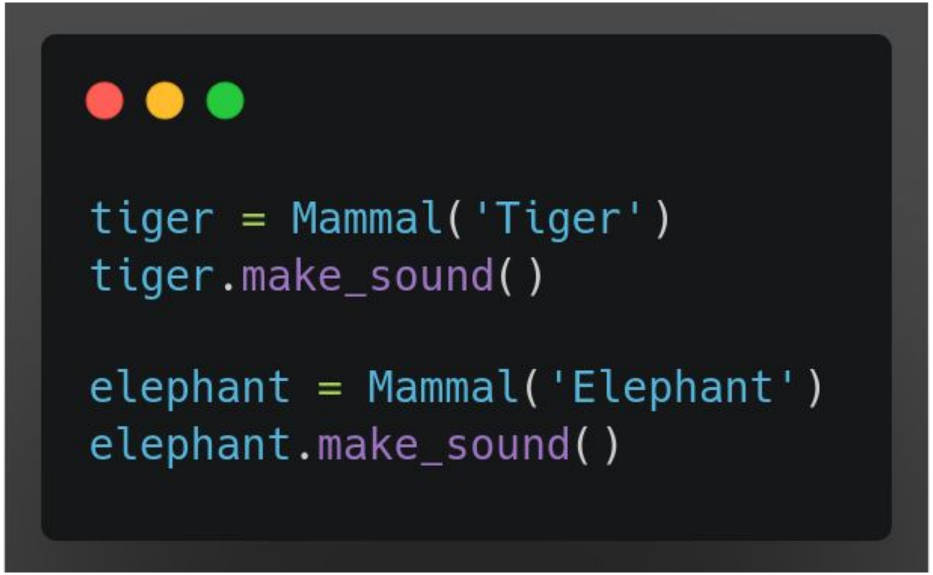
Classes are created by the keyword `class`.



```python
class Mammal:
    def __init__(self, species):
        self.__species = species

    def make_sound(self):
        print('Grrrrr')
```

Object



```
tiger = Mammal('Tiger')
tiger.make_sound()

elephant = Mammal('Elephant')
elephant.make_sound()
```

An object is an instance of a class.

A single integer or single string is an object.

To create/instantiate an object from a class, we call/invoke the class and assign what is returned to a variable.

Benefit of OOP

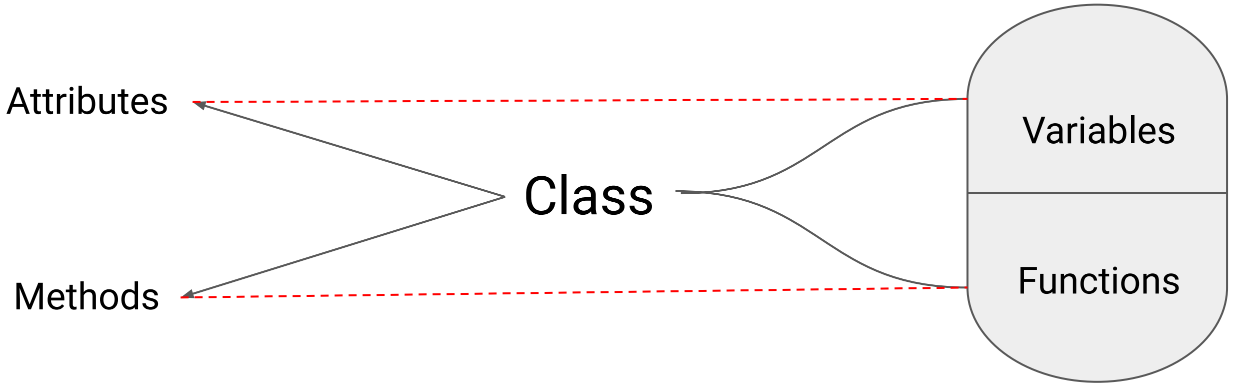
Code that is easy to work with is called ***clean code***.

OOP allows for clean code.

Pillars of OOP

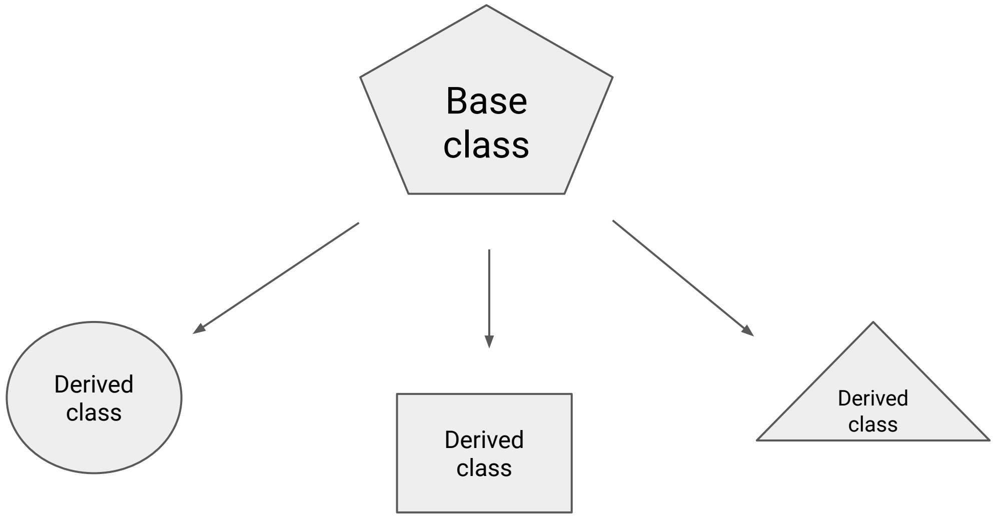
- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

Encapsulation



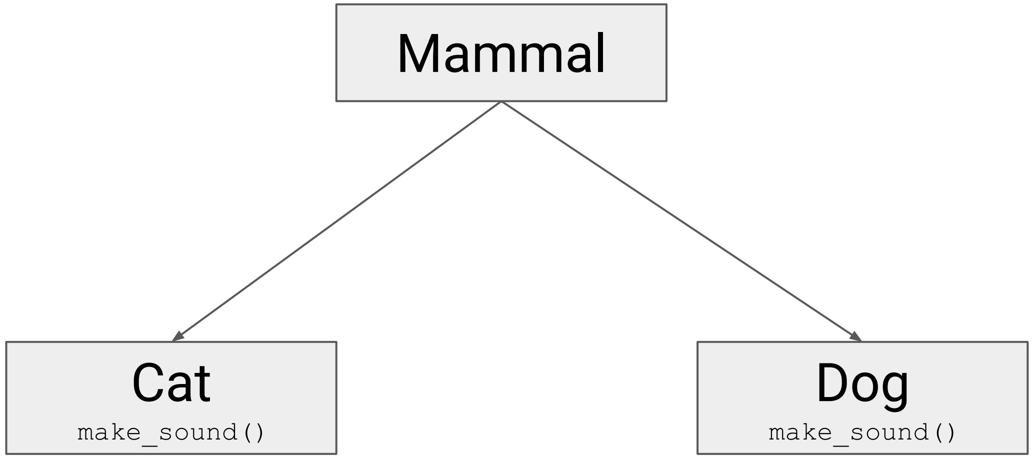
Encapsulation is used to implement ***data hiding*** (Public, protected and private). However, it refers to encasing data and functions into a single unit.

Inheritance



Inheritance refers to creating a new class by utilising the details of an existing class.

Polymorphism

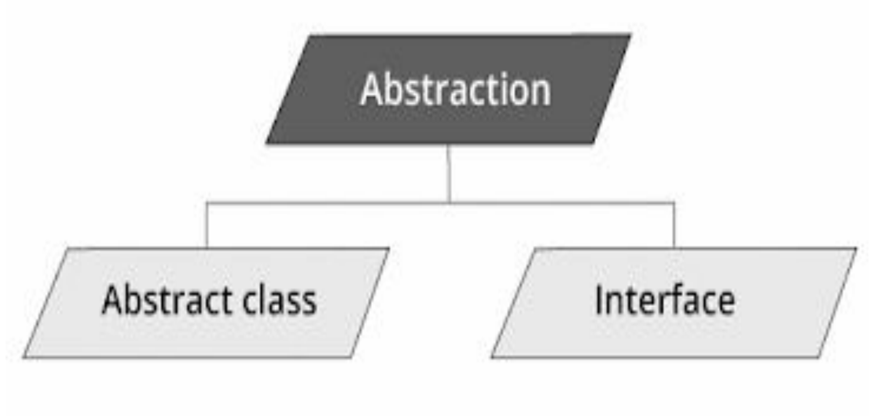


Polymorphism entails the ***same method*** doing ***different things*** for ***different classes***.

Abstraction

Abstraction is hiding unnecessary detail from the user.

It ***hides the implementation details*** while presenting the functionality to the rest of the world.



Conclusion

Herein, you have learned the basic concepts of OOP.

The majority of modern programming languages adhere to OOP concepts, thus the skills you learn here are transferable no matter where your programming career leads you.

Connect with me on here:

LinkedIn: https://www.linkedin.com/in/izundu-dan-ekeh-1b8186196/

Github: https://github.com/Izu-33