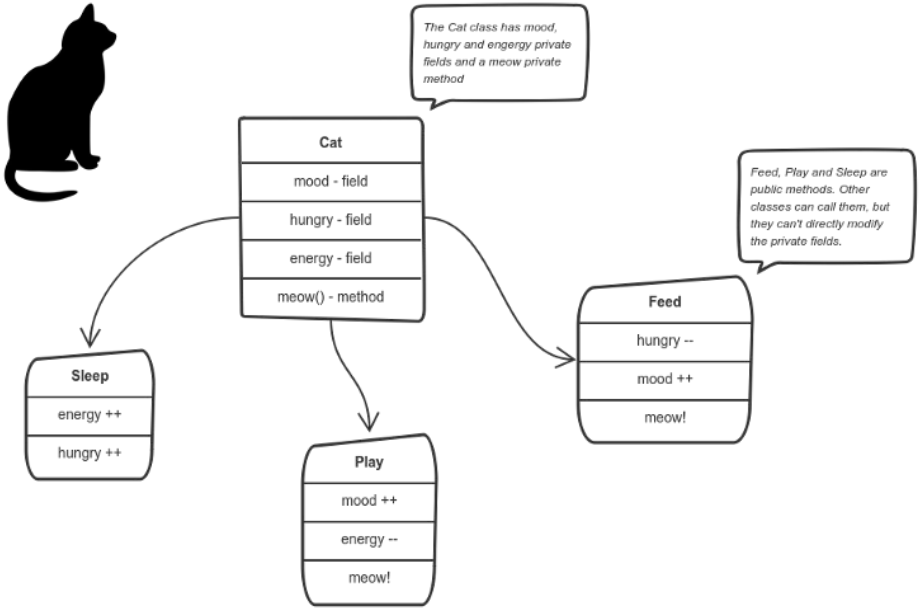
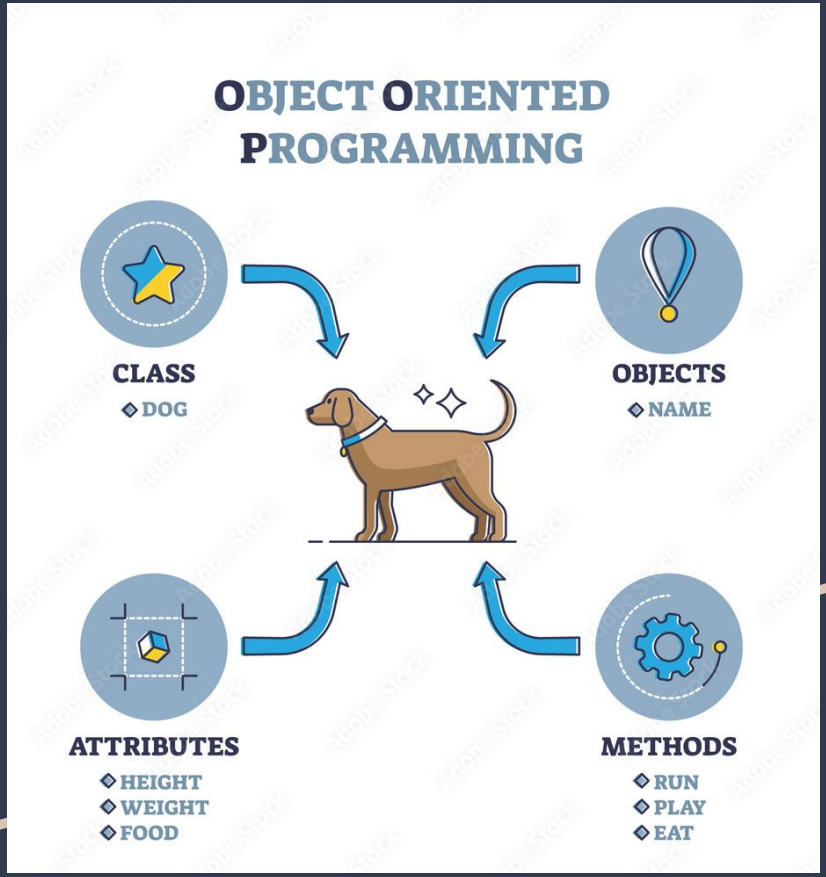
Introduction to Object Oriented Programming

Object-Oriented Programming (OOP)
It is a bottom-up approach
Program is divided into objects
Makes use of Access modifiers ‘public’, private’, protected’
It is more secure
Object can move freely within member functions
It supports inheritance

Object Oriented Programming is a way of computer programming using the idea of “objects” to represents data and methods. It is also, an approach used for creating neat and reusable code instead of a redundant one.

Structure of Object-Oriented Programming





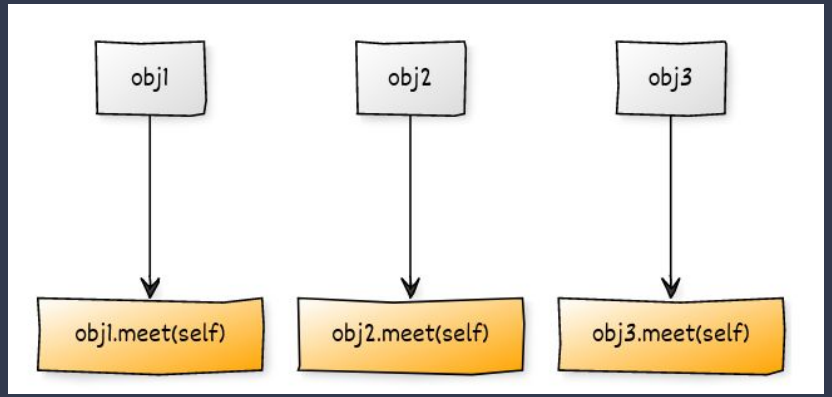
A class is a collection of objects or you can say it is a blueprint of objects defining the common attributes and behavior.

A class contains the blueprints or the prototype from which the objects are being created. It is a logical entity that contains some attributes and methods.

Class is defined under a “Class” Keyword.

A class is a collection of objects.

Attributes are the variables that belong to a class.Attributes are always public and can be accessed using the dot (.) operator. Eg.: Myclass.Myattribute.



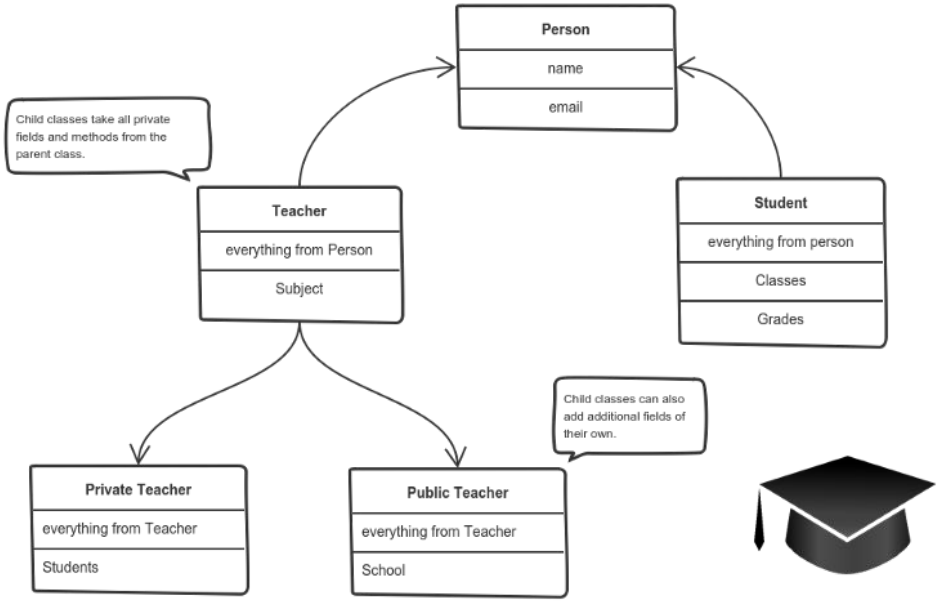
The Python __init__ Method

The __init__ method is similar to constructors in C++ and Java. It is run as soon as an object of a class is instantiated. The method is useful to do any initialization you want to do with your object. Now let us define a class and create some objects using the self and __init__ method.

The Python self

1. Class methods must have an extra first parameter in the method definition. We do not give a value for this parameter when we call the method, Python provides it
2. If we have a method that takes no arguments, then we still have to have one argument.
3. This is similar to this pointer in C++ and this reference in Java.

INHERITENCE



The End