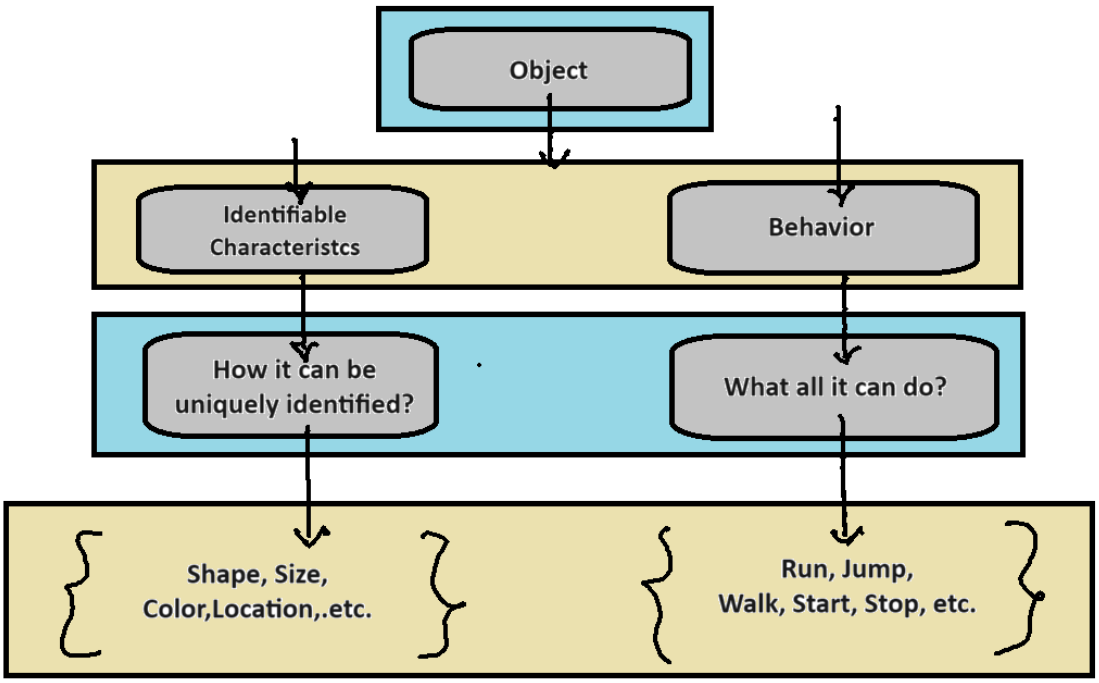
OOPs

Python is a multi-paradigm programming language. It supports different programming approaches. One of the popular approaches to solving a programming problem is by creating objects. This is known as Object-oriented programming(oops).

Object-oriented programming (OOP) is a programming based on the concept of “objects”. The Object contains both data and code: Data in the form of properties (Often known as attributes), and code, in the form of methods (actions the object can perform).



An object has two characteristics:

1. Attributes
2. Behavior

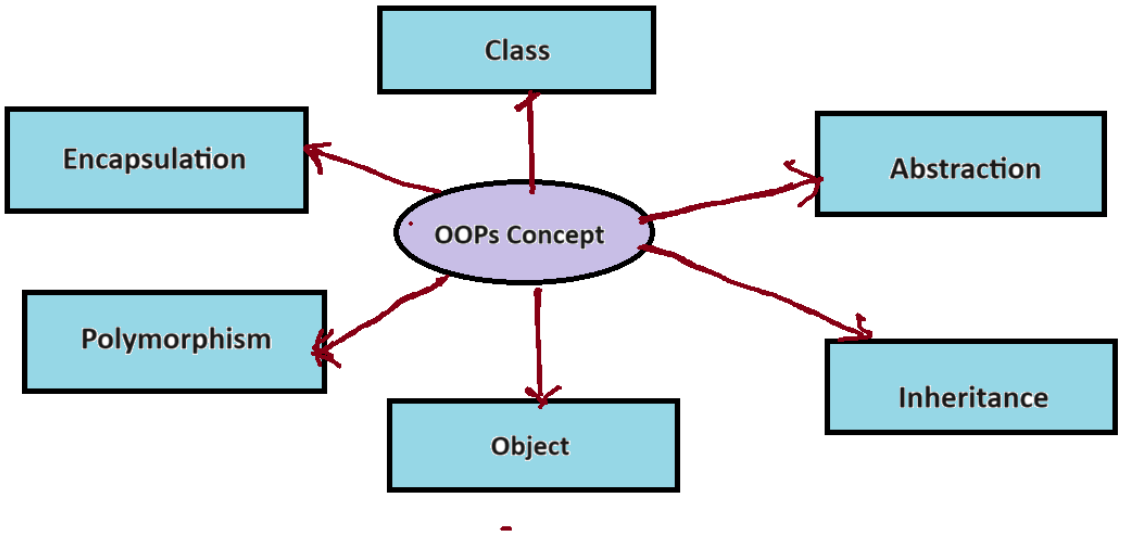
Example:

A parrot is an Object, as it has the following properties:

Name, age, and color as attributes

Singing, dancing, flying as behavior.

Types Of OOPs :



What is Class?

The class is a user-defined data structure that binds the data members and methods into a single unit. Class is a blueprint or code template for object creation. Using a class, you can create as many objects as you want.

Class is defined with the keyword Class

Class Examples: Mercedes, BMW, Skoda, etc.

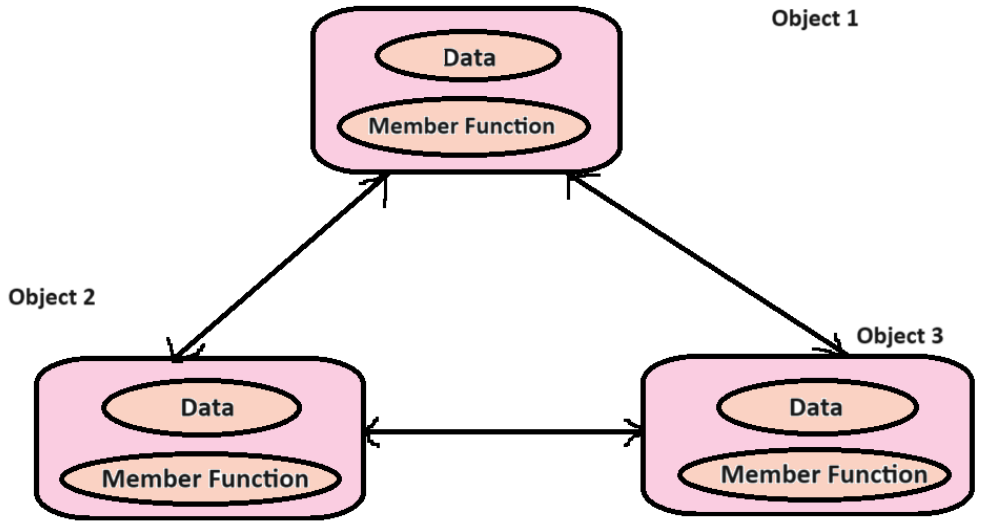
Syntax of class:

```
class ClassName():
    print("Hello!!!")
```

What is Object?
An object is a collection of data(variables) and methods(functions) that act on those data. Similarly, a class is a blueprint for the object.

The object is the instance of a class. The process of creating anobject can be called instantiation. There is no memory allocation until we create its object. The objector instance contains real data or information.

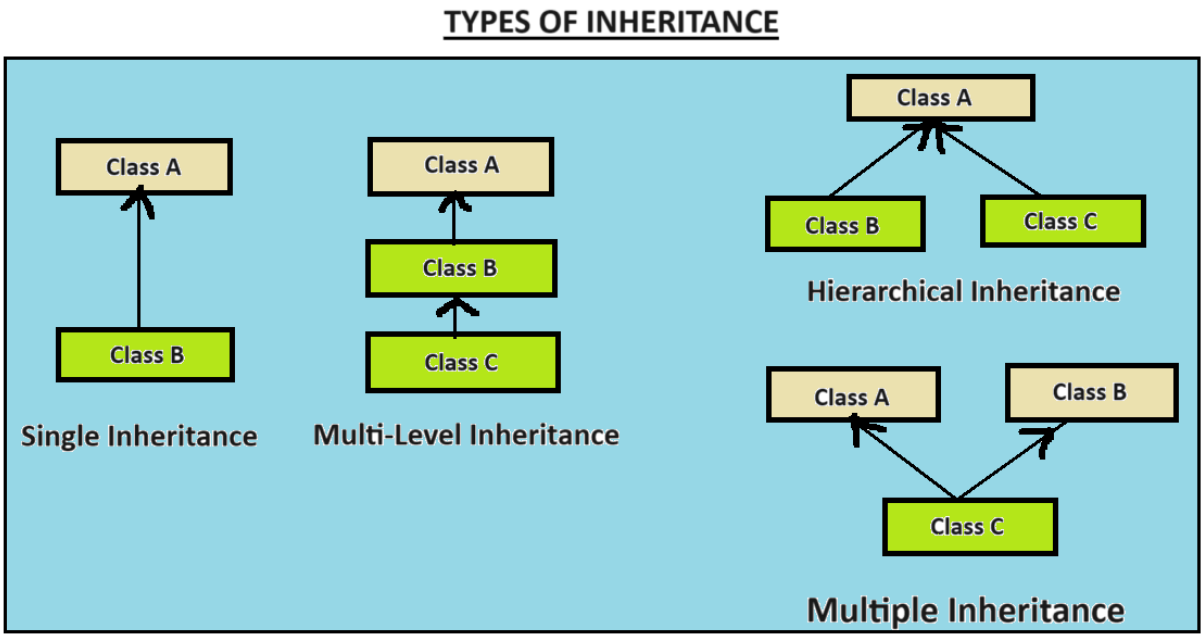
Object Examples: Chair, Bike, Marker, Pen, Car, etc.



<u>Inheritance</u>: Inheritance allows us to define a class that inherits all the methods and properties from another class.

Parent Class is the class being inherited from, also called as base/super class.

Child Class is the class that inherits from another class, also called derived/sub class.



Polymorphism: The literal meaning of polymorphism is the condition of occurence in different forms.

There are two types:

1. Method Overloading
2. Method Overriding

What is Encapsulation?

Encapsulation is a mechanism of wrapping the data(Variables) and code acting on the data(Methods) together as a single unit.

Access Modifiers:

Public: Accessible to the outside world
Protected: Accessible to derived classes
Private: Not accessible to outside