

In []: OOPs- Object Oriented Programming language

object---basis

class

object

inheritance

polymorphism

encapsulation

In []: *# Class*

a **class** serves as a blueprint or template for creating objects.

Definition:

- **class** is the blueprint of the object
- **class** is the template for creating the objects

attribute ---> human, car, color,size,
behavior --->

In [6]:

```
class Dog:
    # class attributes
    species = "german familiaris"
    color = "dark black"
    size = 30

    # constructor
    def __init__(self,name,age):
        # instance attributes
        self.name = name
        self.age = age

    # behaviors
    def bark(self):
        z = 5+ 10 # local variable
        return "barking"

    def running(self):
        return "I can run"

# creating the objects of the class(Dog)

my_dog = Dog("Budy",3)
print(my_dog.species)
print(my_dog.size)
print(my_dog.bark())
print(my_dog.running())
```

german familiaris

30

barking

I can run

In []: *# Questions*

create 5 different class using the name of Human,Animal,Snake,Car,Person