In [ ]:
```
#como programador defino funciones (metodos) que se van a usar repetidamente
#por ejemplo tenemos la funcion print que se definio para usarla repetidamen
#a las funciones tambien se les conoce como METODOS()


#cuando empleo funciones recuerdo 4 cosas

#1 Defino la funcion
#2 Defino si tiene entradas (ARGUMENTOS)
#3 Defino si tiene salidas (RETORNO)
#4 Invoco o uso la funcion (usarla)
```

In [ ]:
```
#Programacion orientada a objetos POO

# CLASES : es un algo que encierra un conjunto de objetos -sirve para catalo
# OBJETOS : un objeto es un elemento (UNA INSTANCIA) que pertenece a una cla
#           el elemento dentro del conjunto
# ATRIBUTOS :los elementos que pueden diferenciar un objeto de otro
# METODOS: funciones -> una cosa que hace algo que necesito para mi objeto

#diferencia entre  objeto e instancia:
#son "lo mismo" un objeto es una instancia de una clase


#constructores: inicializa/crea/hace que exista  el objeto de la clase y le
#               correspondientes a su clase
#SELF: autoparametros  -> puntero -> el codigo entiende sobre quien estan ha
#                        toma los valores del objeto especifico que estoy tr

#JERARQUIA O HERENCIA -> hay un orden, y hay cosas que se pueden heredar
#                        ->hay abuelos, papas, hijos,etc
```

In [11]:

```
# EJEMPLO DE CLASES EN PYTHON (POO)
# PROFE JOSE
# SIC2022

################################## CLASES ###################################

class profesoresSIC: # clase principal o clase padre


    ########## lo primero es un constructor #############
    ########## en el constructor configuro los atributos que pertenecen a mi

    def __init__(self,nombre,apellido,edad,nacionalidad,codigo_gp):
        self.nombre=nombre
        self.apellido=apellido
        self.edad=edad
        self.nacionalidad=nacionalidad
        self.codigo_gp=codigo_gp

    ################################  GETTERS y SETTERS  #########

    #_____________GETTERS_____________________________
    def get_nombre(self):
        return self.nombre
    def get_apellido(self):
        return self.apellido
    def get_edad(self):
        return self.edad
    def get_nacionalidad(self):
        return self.nacionalidad
    def get_codigo_gp(self):
        return self.codigo_gp

    #_____________SETTERS_____________________________

    def set_nombre(self,nombre):
        self.nombre=nombre
    def set_apellido(self,apellido):
        self.apellido=apellido
    def set_edad(self,edad):
        self.edad=edad
    def set_nacionalidad(self,nacionalidad):
        self.nacionalidad=nacionalidad
    def set_codigo_gp(self,codigo_gp):
        self.codigo_gp=codigo_gp

    #_____________otros metodos_____________________________

############################ SUBCLASES ########################

class tutores(profesoresSIC): # clase hijo de profesoresSIC

        ########## lo primero es un constructor #############

        def __init__(self,nombre,apellido,edad,nacionalidad,codigo_gp):
            profesoresSIC.__init__(self,nombre,apellido,edad,nacionalidad,co
```

```
            #reutilizo el constructor de la clase padre
            #porque: LOS METODOS Y ATRIBUTUTOS SE HEREDAN!!!!!!!!!!

################################## MAIN ######################################

profe1 = profesoresSIC("jose","burgos",28,"colombia",1234)
profe2 = profesoresSIC("rafa","puche",30,"venezuela",5678)
profe3 = profesoresSIC("argenis","bouzas",28,"venezuela",9012)
profe4 = profesoresSIC("erick","qwert",25,"panama",3456)

tutor1 = tutores("luis","guitierrez",23,"colombia",1234)
tutor2 = tutores("meidy","asfd",25,"venezuela",5678)

print(profe2.get_nombre())
print(profe2.get_edad())
profe2.set_nombre("giam")
profe2.set_edad(26)
print("_____________")
print(profe2.get_nombre())
print(profe2.get_edad())
print("_____________")
print(tutor1.get_nombre())
```

```
rafa
30

_____________
giam
26

_____________
luis
```

In [ ]:

```
TAREA DE CLASE : PAIR PROGRAMMING

diseñar una clase semestre
    semestre1 = (identificador,numero_de_materias)


    sub clase materias:
        materia1  = ("nombre",intensidadhoraria,codigo,
        materia3
```