

# PROGRAMACIÓN ORIENTADA A OBJETOS

## TALLER No. 4 PYTHON

## PROGRAMACIÓN ORIENTADA A OBJETOS

**Profesor:** Jaime Alberto Guzmán Luna

Contenido del taller:

1. Sobrecarga de métodos y constructores

### Ejercicio 1

- a) Crear una nueva carpeta llamada Taller4 y ábrela en VSCode.
- b) Crea un archivo “main.py” dentro de la carpeta y agrega las siguientes clases y metodos

```
1 class Carro:
2     puertas=4
3     carros=0
4
5     def __init__(self, referencia, peso = 1, marca = "Ford", motor = None):
6         self._referencia = referencia
7         self._peso = peso #Peso en toneladas
8         self._marca = marca
9         self._motor = motor
10        Carro.carros=Carro.carros+1
11
12    def getPeso(self):
13        return self._peso
14
15    def getReferencia(self):
16        return self._referencia
17
18    @staticmethod
19    def carro_mas_pesado(carros):
20        aux_ref=carros[0].getReferencia()
21        aux_peso=carros[0].getPeso()
22        for carro in carros:
23            if (carro.getPeso() > aux_peso):
24                aux_ref=carro.getReferencia()
25                aux_peso=carro.getPeso()
26
27        return aux_ref
28
```

# PROGRAMACIÓN ORIENTADA A OBJETOS

29	if __name__ == "__main__":
30	#carro0=Carro()
31	carro1=Carro("Tracker")
32	carro2=Carro("Sander Stepway", 3, "Renault", "v8")
33	carro3=Carro("Picanto", 2, "Kia")
34	
35	print(Carro.carro_mas_pesado([carro1, carro2, carro3]))
36	

## Responder:

- En la línea 34:
- a) ¿Cuál es el peso del carro1? Explique porque
- b) ¿Cuál es el motor del carro3? Explique porque
- c) ¿Cuál es la marca del carro0 ( En caso de que este no estuviera comentado en la línea 30)? Justifique lo que sucede
- d) ¿Qué imprime la línea 35? Justifique
- Modifique el constructor de la clase de la siguiente manera:

### Línea original

- def \_\_init\_\_(self, referencia, peso = 1, marca = "Ford", motor = None):

### Línea nueva

- def \_\_init\_\_(self, referencia, peso = 4, marca = "Ford", motor = None):

- e) ¿Qué imprime la línea 35 después del cambio? Explique
- f) ¿Cómo modificar el constructor de Carro, para que pueda recibir indefinido número de parámetros
- g) Revisa el siguiente código e identifique los errores que vea en el hasta que logre correrlo, luego de esto responda las siguientes preguntas

1	class Punto:
2	def __init__(x, y=0):
3	self.x = x
4	self.y = y
5	
6	class Circulo:
7	def __init__(self, radio, **kwargs, *args):
8	self.radio = radio
9	self.punto = Punto(**kwargs, *args)
10	
11	
12	cir = Circulo(10,1,1)
13	print(cir.radio)
14	print(cir.punto.x)
15	print(cir.punto.y)
16	

- h) ¿Qué sucede si elimino el argumento \*kwargs de las clases?
- i) ¿Si en la clase Circulo envió mas de 3 atributos esto ocasionaría un error? En caso que si como soluciono este utilizando \*args o \*kwargs

# PROGRAMACIÓN ORIENTADA A OBJETOS

## Ejercicio 2 - GitHub

Enlace entrega y ejercicio: <https://classroom.github.com/a/4zU7kQMn>

### PROBLEMA

Según el código dentro del módulo classroom identifique los errores de compilación y piense la razón por la que ocurren, luego corrijalos.

### IMPORTANTE

Corrija los errores de la forma que mejor lo considere, de modo que el programa funcione correctamente e imprima por pantalla al correr el main, el siguiente resultado. Puede agregar, eliminar o modificar el código. Trate de pensar en la mejor forma de corregirlo. No modifique el main.

### Resultado:

```
Matematicas remoto
Grupo de estudiantes: grupo predeterminado
Grado 12
['Kelly']
['Jaime', 'Pedro', 'Santiago']
['Javier']
3
Grado 1
Grado 6
```