PREGUNTAS PARCIAL 2023 MARZO

1.

```
x = [1,2,3,4,5,6,7,8,9,10]
```
¿Como mostramos el rango de números mayores o iguales que 5?

```
print (x[5:]) o print (x[-5:])
```

2.

```
x = 5.6
y = int (x)
print (y)
```

¿Qué se mostrará por pantalla? → 5

3.

```
x = 10
i = 0
while i < 10 or x:
    print ("Vuelta:"+str(i))
    i +=1
```
¿Cuantas veces se ejecutará el bucle? → Infinitas

4.

```
d = {
    "Total": 6.5,
    "Ejercicios":
    {
        "T1":
        {
            "Estado":"Entregado",
            "Nota": 7.5},
        "T2":
        {
            "Estado":"Entregado",
            "Nota": 5.5},
        "T3":
        {
            "Estado":"Entregado",
            "Nota": None},
    }
}
```
¿Cómo accedemos a la nota de 5.5? →
```
print(d["Ejercicios"]["T2"]["Nota"])
```
5.
```
x = "Hola"
y = x
print (y+str("amigos"))
```
¿Qué mostrará por consola al ejecutarlo? → Holaamigos

6.

```
lst = [5,2,6,7,1]
i = 0
for x in list:
    i = i +1
    if 0 <=x<5:
        continue
    print ("El alumno"+str(i)+"ha sacado un"+str (x))
```

¿Qué mostrará por pantalla? → Error.

7. **¿Qué es un IDE?** → Integrated Development Enviroment
8. **¿Qué diferencias hay entre un lenguaje de programación interpretado y uno compilado?**

   Los lenguajes compilados requieren de un software intermedio (compilador) para traducir las instrucciones a lenguaje máquina. Los interpretados se traducen según se va ejecutando el programa siempre y cuando se disponga del intérprete.
9. **¿Qué tipo de tipado tiene Python?** → Tipado dinámico fuerte
10. **¿Cuál es un nombre de variable válido en Python?** → _nombre=10
11. **¿Qué es un IDE?** → Integrated Development Enviroment
12. **Indica la afirmación correcta.**

    Una función puede devolver o no un valor, los parámetros de entrada pueden ser opcionales y puede contener tantas instrucciones como deseemos.
13. **¿Cómo se elimina el penúltimo elemento de una lista?** → Lst.pop (-2)
14. **Indica que es incorrecto si hablamos de un algortimo. (al contrario pues todas menos esta).**

    Son conjuntos de instrucciones desordenadas
15. **¿Cuáles son las características de Python?**

    Es multiparadigma, multiplataforma e interpretado.
16. **Indica la afirmación correcta.**

    En python x = 10 es lo mismo que x = 10

+preguntas de otros exámenes

17. Dado este código:

```
for x in range (2):
    for y in range (2):
        print (x,y)
```

¿Qué mostrará este código? →

```
0 0
0 1
1 0
1 1
```

18. Dado este código:

```
x = "Programación"
i = 0
while i < len (x):
    print (x[i])
    i+=1
```

¿Cuántas veces se ejecutará el bucle? → 12 veces

19. Dado este código:

```
r = 0
for i in range (10):
    i+=1
    if i%5 == 0 or i>10:
        break
    else:
        r+=1
```

¿Qué valor tendrá r al finalizar el programa? → 4

20. Dado este código:

```
x1 = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21]
x2 = x1
x2 [0]=2
x1[-1]=44
```

¿x1 y x2, serán iguales? → Si

21. Dado el siguiente código:

```
int x = 1.87
print (x)
```

¿Cuál es el resultado al ejecutarlo? → Da error de sintaxis.

22. Dado el siguiente código:

```
x = "Bienvenido"
y = x
print (y + str("amigo"))
```

¿Qué muestra por pantalla? → Bienvenidoamigo

23. Dado el siguiente código:

```
def Func1 (a, b = 10. c = 0):
    if a %2 ==0:
        a*=2
    else:
        b=a+2
    return ((a+b)*c)
Func1(1,10)
```

¿Qué muestra por pantalla? → Nada

24. Dado el siguiente código:

```
x = True
i = 0
while i < 5 or x:
    print ("Vuelta:"+str(i))
    i+=1
```

¿Cuántas veces se ejecutará el bucle? → Infinitas

25. Dado el siguiente código:

```
class Clase1:
    def __init__(self,a):
        self.a = a
        self.b = 100
class Clase2(Clase1):
    def __init__ (self,b):
        super ().__init__ (50)
        self.b = self.a
        self.c = b
objeto = Clase2(100)
```

¿Qué atributos tendrá el objeto y con que valores si ejecutamos el código? → a, b, c = 50,50,100

26. Dado el siguiente código:

```
X = [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1]
```

¿Cómo mostramos el elemento del medio? → print (x[-7:-5])

27. Dado el siguiente código:

```
d1 = {"uno": 1, "dos":2}
d2=d1
d2["tres"]=3
d1["cuatro"]=4
```

¿Qué afirmación es correcta? → d1 y d2 son exactamente iguales.

28. Dado el siguiente código:

```
class Clase1:
    def __init__(a,b):
        self.id = a
        self.nombre = b
def __str__ ():
    return "ID": "+str(self.ID)+"nombre:"+self.nombre
o = Clase1(1,"Eva")
print (o)
```

Si se ejecuta… → Provocará un error debido a que falta la referencia interna a la clase.

29. Dado el siguiente código:

```
x,y,z = 6,3,-3
try:
    print ("Igual"+str (x/(y+z)))
except:
    print ("Error")
```

¿Qué mostrará por pantalla? → Error

30. Dado el siguiente código:

```
class Clase1:
    def __init__(self,a):
        self.a = a
class Clase2(Clase1):
    def __init__ (self,b):
        super ().__init__ (50)
c1 = Clase1(100)
c2 = Clase2 (50)

if type (c1) == type (c2):
    print (1)
elif isinstance (c1,Clase1)==isinstance(c2,Clase1):
    print (2)
else:
    print (3)
```

¿Que mostrará por pantalla? → 2

31. Dado el siguiente Código:

```
x = ["a", 8, 5, "b", 4.3,4,"c",7.5,(1,"d",3,4)]
for i in x:
    if not type (i) == str:
        continue
    print (i)
```

¿Que mostrará por pantalla al ejecutarlo? → a, b, c

32. Dado el siguiente Código:

```
def Func1 (a):
    v = ["a", "e", "i", "o", "u"]
    r = ""
    for i in a:
        if i.lower() in v:
            r+=1
    return r
print (Func1("Programación en Python"))
```

¿Que imprimirá por pantalla? → Nada

33. Dado el siguiente código:

```
x = ["Ricardo", "Rosa", "Sara","Leo", "Ramon"]
z = ["Ana", "Borja", "Kike"]
y = "Kiko"
if y not in x+z:
    x.append(x[-1])
    z.pop
    z.append (y)
```

¿Que afirmación es correcta si ejecutamos el programa? → x tendrá 6 elementos y z tendrá 3 elementos.

34. Dado el siguiente código:

```
def Func1 (*a):
    b = []
    for i in a:
        b.insert (0,i)
    return b
a = Func1 ("Ana", 2, 3, 4, "Pedro")
print (a)
```

¿Que imprimirá por pantalla? → ['Pedro', 4, 3, 2, 'Ana']

35. Dada la siguiente variable:

c= “Programación en Python”

¿Qué mostrará print (c[13:15])? → en

36. Dado el siguiente código:

```
x = "Atipico"
for y in x:
    if y.lower() == "a":
        print (y)
```

¿Que mostrará por pantalla al ejecutar el código? → A

37. ¿Cuál es correcta si hablamos de características de Programación orientada a objetos? → Herencia, polimorfismo, abstracción y encapsulación.

38. ¿Cuál es incorrecta si hablamos de clases en Python?

Para acceder a un atributo dentro de la clase no es necesario utilizar la referencia interna

39. ¿Cómo se llama el constructor de la clase base desde la clase derivada? → super().__init__()

40. El proceso de creación de un objeto de una clase determinada, se denomina… → Instancia

41. Indica la correcta

a)
```
if x < y:
  print("X es menor")
else:
  print("Son iguales")
elif x > y:
  print("Y es menor")
```

b)
```
if x < y:
  print("X es menor")
elseif x > y:
  print("Y es menor")
else:
  print("Son iguales")
```

c)
```
if x < y:
  print("X es menor")
elif x > y:
  print("Y es menor")
elif:
  print("Son iguales")
```

d)
```
if x < y:
  print("X es menor")
elif x > y:
  print("Y es menor")
else:
  print("Son iguales")
```

42. ¿Qué estructuras de control hay en Python?

For, if-else y while

43. Indica la respuesta incorrecta:

a) Una clase no puede heredar mas de dos clases padre.
b) La clase de la que hereda otra clase se llama clase base.
c) La clase que hereda de otra se llama clase derivada.
d) La clase hija puede heredar tanto los métodos como los atributos de la clase padre.

44. ¿Cuál es la sintaxis correcta del operador lógico AND?

If x <=5 and x>0:

45. Indica la respuesta incorrecta para definir una variable.

Es una porción de código reutilizable que se encarga de realizar una determinada tarea