



Control 1

Tiempo: 30 mins

Nombre: _____

1. ¿Cuál es la salida del siguiente código?

```
class test:
    def __init__(self,a):
        self.a=a

    def display(self):
        print(self.a)
test().display()
```

- a) el código no tiene salida.
b) muestra un 0, el valor por defecto de un entero.
c) genera un error, ya que se requiere un argumento en el constructor.
d) genera un error, ya que no se llama a `__init__` luego de crear el objeto.
2. ¿Cuál de los siguientes elementos busca representar una entidad en el *mundo real*, conteniendo su identidad y comportamiento.
- a) un método.
b) un objeto.
c) una clase.
d) un operador.
3. ¿Cuál es la salida del siguiente código (no considere los saltos de línea faltantes)?

```
class A:
    def test(self):
        print("A")
class B(A):
    def test(self):
        print("B")
        super().test()
class C(A):
    def test(self):
        print("C")
```

```
class D(B,C):
    def test(self):
        print("D")
        super().test()
D().test()
```

- a) **D B C.**
- b) D.
- c) D B A.
- d) D B C A.

4. ¿El método `__init__` debe ser llamado inmediatamente después de creado un objeto?

- a) verdadero.
- b) **falso.**
- c) sólo si `__init__` recibe argumentos.
- d) puede ser llamado después, pero antes de cualquier otro método de la clase.

5. ¿Cuál de las siguientes descripciones aplica de mejor manera al mecanismo de herencia?

- a) **la habilidad de derivar miembros de otra clase, a partir de su propia definición.**
- b) un medio para empaquetar variables y métodos, de manera de restringir su acceso.
- c) permite la implementación de software de manera simple y elegante.
- d) un mecanismo para evitar duplicación de código.

6. ¿Si **B** es una subclase de **A**, como se puede llamar el método `__init__` de **A** a través de **B**?

- a) `A.__init__(self).`
- b) `B.__init__(self).`
- c) **`A.__init__(B).`**
- d) `B.__init__(A).`

7. ¿Cuál es la salida del siguiente código?

```
class A:
    pass
class B(A):
    pass
obj=B()
print(isinstance(B,A))
```

- a) True.
- b) **False.**
- c) genera un error por sintaxis inválida.
- d) genera un error por que este método no puede aplicarse a la clase A.

8. ¿Cuál es el uso del *duck typing*?

- a) Aumentar las restricciones a las clases que pueden ser pasadas a los métodos, para minimizar problemas.
- b) Hacer al código más pequeño.
- c) **Disminuir las las restricciones a las clases que pueden ser pasadas a los métodos, para aumentar la flexibilidad.**
- d) Permitir un mecanismo de herencia más controlado.