

**TALLER No. 2 PYTHON**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

1. ¿Cuántas clases se están definiendo en este ejercicio?
2. ¿Para qué sirve la línea de código **if \_\_ name = “ \_main\_ ”**:?
3. ¿Qué sucede si retiro la línea de la pregunta anterior en nuestro código?,¿Este sigue corriendo o hay error? Explique en ambos casos.
4. ¿Cuántos objetos de la clase **Apostador** se están creando?
5. ¿Cuáles objetos de la clase **Apostador** se están creando?
6. ¿A quién está haciendo referencia la variable **self** de la línea **15** de la clase **Apostador**

cuando se ejecuta el programa principal*?*

1. ¿Cuántos objetos de la clase **Loteria** se están creando?
   * En la línea **4** del **main.py** cambiar el apostador1.deposit(500) por apostador1.deposit(300)
2. ¿Qué imprimiría el código por parte del **apostador1**?
   * En la línea **10** del **main.py** cambiar el apostador2.deposit(500) por apostador2.deposit(400)
3. ¿Qué imprimiría el código por parte del **apostador2**?
4. ¿Cuáles atributos de la clase **Lotería** están haciendo referencia a objetos?
5. ¿Cuáles atributos de la clase **Lotería** están haciendo referencia a tipos primitivos?
6. ¿Complete las siguientes líneas para que en la clase **Loteria**, se implemente el método de clase changeProbability?

-

* + def changeProbability(\_ , nprobability):

### \_ .probability = nprobability

1. ¿Cómo sería la línea de código para llamar el método changeProbability?
2. ¿Es correcto si en el método changeProbability que se creó, cambiar lo siguiente? Explique:

**Linea Original**

### cls.probability = nprobability

**Linea Nueva**

### Loteria.probability = nprobability

1. ¿Cuántos métodos tiene la clase **Loteria** después de agregarle el nuevo método?
2. ¿Si el **apostador1** gana el **apostador2** también? Explique porque pasa en caso de ser sí o no
3. ¿Qué sucede si decido cambiar el atributo de clase probability a una constante? ¿Se considera correcto el uso del método changeProbability teniendo en cuanta este nuevo cambio?
4. ¿Cuál es el tipo de retorno de los métodos gain() y commission() de la clase

**ComisionJuegoEspectaculos**?

1. ¿A quién está haciendo referencia la variable **self** de la línea **18** de la clase **Loteria** cuando se ejecuta el programa principal*?* ¿Podría omitirse el uso de la variable **self** en este caso?
2. ¿En la línea **15** de la clase apostador vemos como la clase recibe dos parámetros (value, self) especificar cuál de estos pasa por valor y cual por referencia y por qué?