1. **¿Cuántas clases se están definiendo en este ejercicio?**

**3, loteria, apostador, y cmoisionjuegosespectaculos**

**b) ¿Para qué sirve la línea de código if \_\_ name = “ \_main\_ ”:?**

**Verifica si el programa que esta corriendo es el programa main, en caso de que no sea, no se ejecutara**

**c) ¿Qué sucede si retiro la línea de la pregunta anterior en nuestro código?, ¿Este sigue  
corriendo o hay error? Explique en ambos casos.**

**Si se ejecuta de forma directa el archivo.py, correra de mantera normal, pero en caso que ese programa se importe como modulo, podría generar problema al ejecutarse cuando se importe**

**d) ¿Cuántos objetos de la clase Apostador se están creando?**

**e) ¿Cuáles objetos de la clase Apostador se están creando?**

**2, juan y Ricardo**

**apostador1 = Apostador(1, "Juan", 302, "j@gmail.com")**

**apostador2 = Apostador(2, "Ricardo", 548, "r@gmail.com")**

**f) ¿A quién está haciendo referencia la variable self de la línea 15 de la clase Apostador  
cuando se ejecuta el programa principal?**

**El método play se ejecuta 2 veces, cada vez hace referencia al objeto que la invoco, por ejemplo cuando juan juega, hace referencia ajuan y el mismo caso con Ricardo**

**g) ¿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)**

**solo se crearía 1, porque el apostador 1 no tiene suficiente dinero para apostar**

**h) ¿Qué imprimiría el código por parte del apostador1?  
- En la línea 10 del main.py cambiar el apostador2.deposit(500)  
por apostador2.deposit(400)**

**no afectaría ya que esta modificando un atributo del jugador 2**

**i) ¿Qué imprimiría el código por parte del apostador2?**

**jugaría de la misma manera, solo que tendría 100 menos en la billetera, pero si cumple la cantidad para apostar**

**j) ¿Cuáles atributos de la clase Lotería están haciendo referencia a objetos?**

**Dentro de de la clase se hace referencia al apostador**

**k) ¿Cuáles atributos de la clase Lotería están haciendo referencia a tipos primitivos?**

**El atributo valor**

**l) ¿Complete las siguientes líneas para que en la clase Loteria, se implemente el método de clase changeProbability?  
-@classmethod  
- def changeProbability(\_\_, nprobability):  
- \_cls .probability = nprobability  
m) ¿Cómo sería la línea de código para llamar el método changeProbability?**

**Loteria.changeProbability(0.3)**

**n) ¿Es correcto si en el método changeProbability que se creó, cambiar lo siguiente?  
Explique:**

**- cls.probability = nprobability  
Línea Nueva  
- Loteria.probability = nprobability**

**Funcionaria igual, pero se podrían presentar problemas a la hora de heredar la clase**

**o) ¿Cuántos métodos tiene la clase Loteria después de agregarle el nuevo método?**

**5, el init, paymoney, recievemoney, playgame, y el nuevo, change probability**

**p) ¿Si el apostador1 gana el apostador2 también? Explique por qué pasa en caso de ser sí o  
no**

**no, ya que cada tiro de la loteria tiene un resultado aleatorio e independiente**

**q) ¿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?**

**Si se convierte en una constante eso significa que no puede modificarse con un método en un futuro, por lo que el nuevo método no funcionaria**

**r) ¿Cuál es el tipo de retorno de los métodos gain() y commission() de la clase  
ComisionJuegoEspectaculos?**

**Tanto gain como comisión devuelven números de tipo float**

**s) ¿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?**

**No, no se puede omitir esta variable ya que es nescesaria para que se cobre la comisión de juegos, a el objeto que esta ejecutando la función, en este caso, el jugador**

**t) ¿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 cuál por referencia y por qué?**

**Value se pasa por valor ya que es un dato primitvo, el cual se copia para manipularse en otro estado de dirección de memoria, en cambio el atributo self apunta a la misma instancia, un objeto**