

1. Si deseo modificar el mensaje del método ruido al crear un objeto Pajaro sin alterar la clase Animal ¿Qué debo agregarle al código? (Por ejemplo, al llamar el método ruido imprima, cantar y silbar)  
R// Debemos de sobrescribir este método dentro de la clase Pajaro.  
El ejemplo seria el siguiente:  

```
def ruido(self):  
    return "cantar y silbar"
```
2. Si se crea una nueva clase Pez, y no se definen nuevos métodos, constructor y atributos. ¿Qué constructor tendrá esta clase, qué argumentos recibe? ¿Qué otros métodos y atributos tendrán estos mismos?  
R// Esta clase tendrá el constructor por defecto, sin embargo tendra todos los metodos y recibirá los mismo atributos que el inicializador de su padre (Animal) Puesto que en Python el inicializador es un método, por lo que este se hereda.
3. ¿Qué ocurre con el atributo nombre y edad de la clase SerVivo, al momento de definirse en la Clase Animal? ¿Cómo cambiaría el código del constructor para que estos atributos sean el mismo?  
R// En la clase Animal estos atributos se están sobrescribiendo, ya que se están definiendo nuevamente en el constructor. La forma en la que cambiaria el código seria la siguiente:  

```
def __init__(self, nombre, edad, raza):  
    super().__init__(nombre, edad)  
    self._raza = raza  
    Animal._totalCreados += 1
```
4. En la clase Animal se sobrescribieron los métodos setNombre y getNombre, ¿Como modificaría estos métodos para que su funcionamiento no oculte algún valor de la clase padre? ¿Podría plantearse esta solución usando super()?  
R// Si, la modificación seria la siguiente:  

```
def setNombre(self, nombre):  
    super().setNombre(nombre)  
  
def getNombre(self):  
    return super().getNombre()
```
5. El atributo totalCreados de la clase SerVivo ¿es heredado por las demás clases? ¿En este caso ocurre ocultación de los atributos al definirlo de nuevo en las clases hijas?  
R// Si, este es un atributo de clase heredable. La ocultación efectivamente ocurre, ya que al volver a declarar el atributo, este se sobrescribe y deja de pertenecer a la clase de la originalmente era.

6. ¿Los métodos getTotalCreados sobrescriben al metodo de su padre?  
R// Si, ya que al ser de clase esto pasa a tratarse por medio de la ligadura estatica, la cual hará que se ejecute el método declarado en la clase del objeto que lo llamo.
7. ¿Qué métodos hereda una clase que hereda de la clase Persona?  
R// Este heredara tanto los metodos de SerVivo como los metodos de Persona, es decir:  
[setNombre, getNombre, setEdad, getEdad, getTotalCreados, aduenarAnimal]
8. ¿Qué tipo de objetos podrían pasársele al método aduenarAnimal de la clase Persona? ¿Se le puede pasar un objeto serVivo?  
R// A este método solo se le podrían pasar subclases de Animal. serVivo no seria adecuado debido a que este no tiene el método ruido, el cual es uno que solo tienen los animales.
9. ¿Cómo podría obtener la cantidad de seres vivos que se creen (objetos creados)?  
R// Utilizando el método getTotalCreados de la clase Animal, y no sobrescribiendo este método en las demás clases.
10. Si se define el siguiente método en la clase Perro:  
def getRaza(self, tipo):  
return self.\_raza + " " + tipo  
¿Qué ocurre con el método getRaza de la clase Animal? ¿Este método se sobrescribe o se sobrecarga?  
Este método se sobrescribe debido a que en Python no existe la sobrecarga, por lo que al detectar que este nuevo método tiene el mismo nombre, adopta una nueva definición de este para la clase.