

1. Si deseo modificar el mensaje del método pitar al crear un objeto moto sin alterar la clase Auto, ¿qué debo agregarle al código? (Por ejemplo, al llamar el método pitar imprima: Las motos no pitan).

Respuesta: Se debe sobrescribir en método pitar() en la clase moto utilizando el decorador @Override de la siguiente manera:

```
@Override  
public void pitar() {  
    System.out.println("Las motos no pitan");  
}
```

2. Suponga que se agrega una nueva clase al código, `class Motoneta`, y esta hereda de la clase `Moto`, ¿evidencia algún problema? ¿Por qué?

Respuesta: La nueva clase `Motoneta` puede heredar correctamente a la clase `Moto`, sin embargo se debe definir bien su constructor o utilizar el ya definido en la clase `Moto` con `super`

3. Suponga que se definió el método: `public void arrancar() { System.out.println("Arrancando"); }` en la clase `Moto`, ¿es posible sobrescribir el método? ¿Por qué?

Respuesta: Si se puede sobrescribir, por que el método es publico, no es final y no es estático

4. En la línea 13 de la clase `moto`, ¿Por qué puedo utilizar el método `pitar`?

Respuesta: lo puede utilizar por que el método `pitar` es heredado por `Moto` desde la clase `Auto`, además el método es publico

5. Haciendo una pequeña modificación a la clase `Auto` y utilizando la variable `num_autos`, sin modificar su modificador de acceso, ¿cómo puedo obtener el número de autos creados desde la clase `ObjTaller5H`?

Respuesta: para poder acceder al numero de autos creados sin modificar su acceso, se debe crear un método publico que retorne su valor, de la siguiente manera y Para obtener el numero de autos desde la clase ObjTaller5H, basta con hacer un llamado al método con la siguiente línea por ejemplo:

System.out.println(Auto.getNumAutos())

```
public static int getNumAutos() {  
    return num_autos;  
}
```

6. En la línea 7 de la clase ObjTaller5H, ¿Por qué no puedo utilizar el método adelantar, si este fue heredado?

Respuesta: Por que el método posee el modificador por defecto (De paquete) entonces solo puede ser heredado/Utilizado desde clases del mismo paquete

7. En la línea 8, ¿por qué es posible utilizar el método arrancar si la clase Bus no lo define?

Respuesta: Por que lo hereda del método publico arrancar() de la clase Auto

8. En la línea 9 de la clase ObjTaller5H, ¿por qué no puedo utilizar el método pitar, si este fue heredado?

Respuesta: El método pitar si se puede utilizar por que es un método publico que es heredado desde la clase auto

9. En la línea 10 de la clase ObjTaller5H, ¿qué imprime el método getVelocidad()? ¿Por qué?

Respuesta: Imprime 10, por que el atributo velocidad de la clase Auto es no estático y no se sobrescribe en la clase Auto por que el método getVelocidad de la clase auto lo mantiene en su valor establecido en su clase con el .this

10. Si quisiera obtener el valor de la placa de las clases Moto y Bus, además de su modelo y capacidad respectivamente, ¿Que debo agregar al código?

Respuesta: Hacer los respectivos métodos get en cada clase, y si quisiéramos imprimirlo se debe añadir en la clase ObjTaller5H de la siguiente manera (ejemplo con la capacidad del bus):

```
public int getCapacidad() {  
    return this.capacidad;  
}
```

```
System.out.println("Capacidad: " + bus.getCapacidad());
```