

1.

```
public Moto(String placa,String modelo){  
    this.placa = placa;  
    this.modelo=modelo;  
    this.pitar= "Las motos no pitar";  
    this.pitar();  
}
```

Como la variable pitar es pública, desde la clase hija moto se le puede cambiar el valor a la variable pitar para que imprima el nuevo mensaje

2. Si se hereda de moto, no se tendrán los atributos placa y modelo de la clase moto ya que estos son privados.
3. Sí es posible ya que el método hijo prima ante el método del hijo padre, es decir, se puede sobrescribir.
4. Porque es un método que está siendo heredado de la clase auto y este es público.
- 5.

```
System.out.println(Auto.num_autos);
```

Como num_autos es una variable estática, se puede acceder a ella desde la clase sin necesidad de crear un objeto.

6. No se puede utilizar el método porque este no tiene un nivel de acceso explícito, entonces se toma el nivel de acceso por defecto que es privado.
7. Porque el método se está heredando de la clase Auto.
8. El método pitar sí se puede utilizar ya que es heredado de la clase Auto y es un método público.
9. El método imprime 10 que es el valor de la clase auto porque en la clase moto como este método no está definido entonces usa el método de la clase Auto el cual hace referencia a la variable de la clase Auto.
10. Se debe agregar el método get para acceder a los atributos privados

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

```
public String getPlaca() {  
    return this.placa;  
}  
  
public String getModelo() {  
    return this.modelo;  
}
```