

Respuestas:

1. Primero, puedo definir un atributo String en la clase Moto de la siguiente forma:

```
public String pitar = "Esta moto tiene su pito malo";
```

Además, se define un método “pitar” en la clase Moto que tenga una firma idéntica al método “pitar” que está en la clase Auto, esto es para que haya un override del método cuando se ejecute el método Main en la clase ObjTaller5H

```
public void pitar(){  
    System.out.println(this.pitar);  
}
```

2. No hay un constructor por defecto en la clase Moto ni un constructor inicializado en la clase que se acaba de crear, esto se debe a que al extender la clase, hace un llamado por defecto a la clase Padre (en este caso, a “Moto”).

3. No, no se puede reescribir el método como está estructurado, ya que el main no lo ejecuta, solo ejecuta un método arrancar para la clase Bus. Para cambiar esto, se puede agregar la siguiente línea adicional en el main.

```
moto.arrancar();
```

Esta línea hace que el método si se reescriba, pero como estaba estructurado, no hay reescritura

4. Porque se está heredando desde la clase padre Auto. Este método es público, por lo tanto, es accesible para cualquier paquete en el proyecto.

5. En primer lugar, se puede crear un método en clase Auto que imprima el número de autos creados usando la variable num_autos. Cabe recalcar que este método será definido como uno estático puesto que la variable también está definida como una variable de tipo estático.

```
public static void numAutos() {  
    System.out.println(num_autos);  
}
```

Para ejecutarlo en el main, basta con poner la siguiente línea en el main de la clase ObjTaller5H, que hará un call a la clase Auto y su método numAutos. Este nos arroja el resultado de “2”, que pertenece a un objeto Bus y un objeto Moto creados. En Eclipse, esto de manera automática creará un import para el paquete1 de la clase Auto

```
Auto.numAutos();
```

6. No, no es posible usar el método arrancar, puesto que, la línea de código en Auto no tiene definido un “public”, lo que la hace privada para cualquier otro paquete. Auto pertenece al

paquete1, mientras que tanto Moto como ObjTaller5H pertenecen al paquete2. Así, el main no puede ver este método.

```
void arrancar(){  
    System.out.println("Encendido");  
}
```

7. Porque el método arrancar es heredado desde la clase Auto, ya que la clase Bus es una clase hija de la clase Auto. Bus puede heredar los métodos de sus clases padre y usarlos.

8. El código se ejecuta de manera correcta en el código proporcionado por GitHub. Sea por Eclipse o por VS Code.

9. Esta línea no imprime nada. Esto se debe a que el método getVelocidad no está llamado de manera explícita. El atributo velocidad en realidad nunca es accedido de manera directa en el main, además de que el return del método getVelocidad nunca es usado

10. Primero, debo definir un método toString en ambas clases que me haga un retorno de los valores. Esto se verá de la siguiente manera (cada uno en su respectiva clase).

```
public String toString() {  
    return "La moto tiene placa " + placa + ", es una moto modelo " + modelo +  
    "y tiene una velocidad de: " + velocidad;  
}
```

```
public String toString() {  
    return "El bus tiene una placa " + placa + " y una capacidad de: " +  
    capacidad;  
}
```

Después, se va a agregar dos prints en el main:

```
System.out.println(moto);  
System.out.println(bus);
```

Así, obtendremos los siguientes prints al final del código.

```
La moto tiene placa XYZ123, es una moto modelo 2019 y tiene una velocidad de: 30  
El bus tiene una placa ABC345 y una capacidad de: 20
```