1. Suppose that you are working at a vehicle manufacturer. Your job is to write software related to the company's new environment-friendly product – an electric scooter. To do this, you wrote the following `Vehicle` class and its Java subclasses .

```
class Vehicle { ... }

class ElectricVehicle extends Vehicle { ... }

class Scooter extends ElectricVehicle { ... }
```

In the `ElectricVehicle` class, you chose to implement a function with the following method signature:

```
class ElectricVehicle extends Vehicle {
    public ElectricVehicle convert(ElectricVehicle ev);
}
```

Now, you need to override this method in the `Scooter` class. Which of the following method signatures would be valid if your goal was to implement a true subtype? Fill in the corresponding column.

| | True Subtype? | Java Subtype? |
|---|---|---|
| `ElectricVehicle convert(Vehicle ev);` | x | overload |
| `Vehicle convert(ElectricVehicle ev);` | | |
| `Scooter convert(Vehicle ev);` | x | override |
| `Scooter convert(ElectricVehicle ev);` | x | override |
| `ElectricVehicle convert(Scooter ev);` | | |
| `Scooter convert(Object ev);` | x | overload |
| `Object convert(ElectricVehicle ev);` | | |

In the previous problem, you identified some method signatures which would be valid in a true subtype. Of these signatures, which would Java override and which would they overload?