Previous Unit 8 of 14 Next

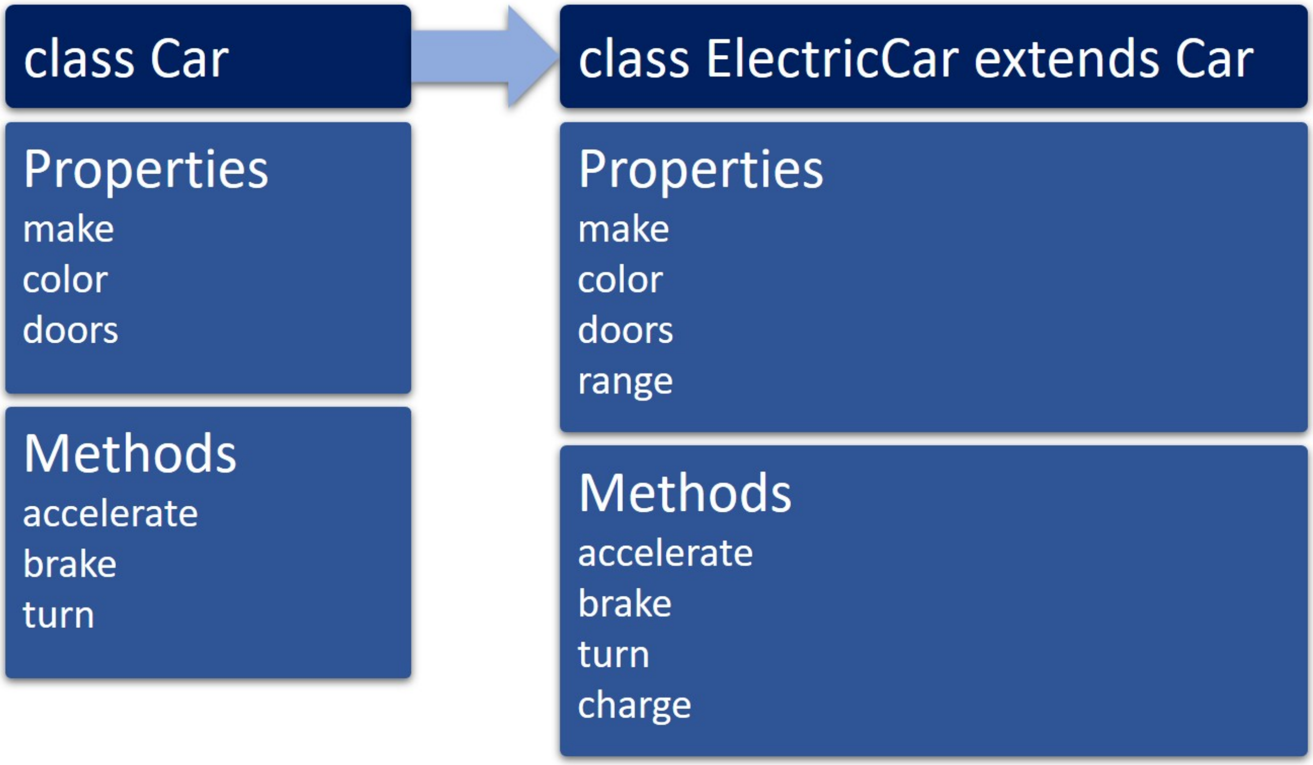
✓ 100 XP

Extend a class using inheritance

2 minutes

Inheritance enables you to establish relationships and build hierarchies of classes in object composition.

For example, you can `extend` the `Car` class to create a new class called `ElectricCar`. The `ElectricCar` class will **inherit** properties and methods of the `Car` class but can also have its own unique attributes and behaviors, like `range` and `charge`. So, by extending the `Car` class, you can create new classes that reuse the code in the `Car` class and then build on it.



The Car class includes the properties make, color and, doors and the methods accelerate, brake, and turn. When the ElectricCar class extends Car, it includes all of the properties and methods of Car, plus a new property called range and a new method called charge.

`ElectricCar` is a **subclass** that uses the `extends` keyword to derive from the Car **base class**. (Base classes are also called **superclasses** or **parent classes**.) Because `ElectricCar` extends the functionality from `Car`, you can create an instance of `ElectricCar` that can `accelerate` , `brake`, and `turn`. If you needed to make changes to the code in the base class, you only need to change it in the `Car` class and then all subclasses of `Car` will inherit those changes.

Some reasons to use inheritance include:

- Code reusability. You can develop once and reuse it in many places. This also helps you avoid redundancy in your code.
- You can use one base to derive any number of subclasses in a hierarchy. For example, the subclasses in the `Car` hierarchy could also include an `SUV` class or a `Convertible` class.
- Instead of having to make code changes in many different classes that have similar functionality, you just need to make the changes once in the base class.

Override a method

When a derived class has a different definition for one of the member functions of the base class, the base function is said to be **overridden**. Overriding is what happens when you create a function in a subclass with the same name as the function in the base class but, it has different functionality.

For example, assume that electric cars use a different type of braking system than traditional cars called regenerative braking. So, you may want to override the `brake` method in the `Car` base class with a method that is specialized for the `ElectricCar` subclass.

Next unit: Exercise - Extend a class

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆