Superclass and Subclass

In inheritance, one class inherits the members of another class. The class that inherits is called a *subclass* or *derived class*. The other class is called a *superclass* or *base class*.

In our car example, `Sedan` and `Truck` are subclasses (or derived classes). They will inherit members from a new class called `Vehicle`, which is the superclass (or base class).

Before using inheritance, both classes had:

- `Wheels`, `LicensePlate`, and `Speed` properties
- `Honk()`, `SpeedUp()`, and `SlowDown()` methods
- Similar constructors

We can pull these out of both classes and put it in a `Vehicle` class. `Sedan` and `Truck` will still have access to those members, but we only need to write them in one place.

By the way, this inheritance hierarchy can extend either way: a new `PickupTruck` class could inherit from `Truck`, which inherits from `Vehicle`, which inherits from a new `Machine` class. The only rule is that a class can only inherit from one base class, e.g. `Vehicle` can't inherit from `Machine` and `Contraption`.

☑**Instructions**

Take a look at this diagram representing inheritance.

- `Sedan` and `Truck` inherit from `Vehicle`
- Members in black font are defined in that class
- Members in grey font have been inherited from a superclass

For example, `Wheels` is defined in the `Vehicle` class and inherited by `Sedan` and `Truck`. `Truck()` is defined only in the `Truck` class.