Review

Well done! You learned a lot very quickly, so let's do a review:

Inheritance is a way to avoid duplication across multiple classes.

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*.

We can access a superclass' members using `base`. This is very useful when calling the superclass' constructor.

We can restrict access to a superclass and its subclasses using `protected`.

We can override a superclass member using `virtual` and `override`.

We can make a member in a superclass without defining its implementation using `abstract`. This is useful if every subclass' implementation will be different.

☑Instructions

The completed code is provided for you here.

Make sure you are comfortable with inheritance before you move on from this lesson. Here are a few questions to test yourself:

In **Program.cs**, `Bicycle.Describe()` is called. Find the definition for that method in the `Bicycle` class, then find the abstract definition of that method in `Vehicle`.

In **Program.cs**, a `Sedan` is instantiated. Find the constructor definition in the `Sedan` class. What happens when that constructor calls `base()`?

In **Bicycle.cs**, `SpeedUp()` is defined. How is it different from `SpeedUp()` in the `Vehicle` class?