

# Implementing an Interface Again

We've completed a `Sedan` class that satisfies both car designers and highway patrol: it can be constructed and change speed, and it implements the `IAutomobile` interface.

But sedans aren't the only automobile on the road. There can be multiple classes that implement an interface.

For example, we can create a `Truck` class that also implements the interface.

This is where we start to see the power of interfaces. Even though `Sedan` and `Truck` are different types, we can assume that they behave similarly because they share an interface. Car designers can build different vehicle classes, but the highway patrol can treat them all the same.

## ☒ Instructions

1.

In **Truck.cs**, create an empty `Truck` class that implements the `IAutomobile` interface.

Hint



Here's an example class that implements `IFakeable`:

```
class Faker : IFakeable
{
}
```

2.

You should see the `error CS0535` telling you that the `Truck` needs to implement the interface! Implement the interface by adding the three properties and one method defined in `IAutomobile`, which you can check in **IAutomobile.cs**.

Hint



If you're not sure what to write next in the `Truck` class, check **`IAutomobile.cs`**.

When you are defining `Honk()`, use `Console.WriteLine()` to print a honking noise to the console.