Implementing an Interface

Our interface is complete! Pretty easy, right?

As we design our automobile-like classes, we'll need to implement this `IAutomobile` interface. In C#, we must first clearly announce that a class implements an interface using the semicolon syntax:

```
class Sedan : IAutomobile
{
}
```

This empty `Sedan` class "promises" to implement the `IAutomobile` interface. In other words, it must have the properties and methods the highway patrol asked for (`Speed`, `LicensePlate`, `Wheels`, and `Honk()`).

If we don't, we get a type error like this:

```
error CS0535: Sedan does not implement interface member
'IAutomobile.LicensePlate'
```

To fix this we'll need to define the members in the interface:

```
class Sedan : IAutomobile
{
  public string LicensePlate
  { get; }

  // and so on...
}
```

Remember that these members must be `public`. How else will the highway patrol be able to access them?

☑Instructions

1.

In **Sedan.cs**, create an empty `Sedan` class that implements the `IAutomobile` interface. Use colon (`:`) notation.

Hint

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

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

2.

You should see the `error CS0535` telling you that the Sedan 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 `Sedan` class, check **IAutomobile.cs**.

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