Finish Truck Class

The car designers have asked that trucks act a bit differently from sedans. Trucks need a new property called `Weight`. Whenever a truck is constructed, its number of wheels will depend on its weight. For example, a heavier truck might need 12 instead of 8 wheels to support itself.

Just like sedans, trucks will also `SpeedUp()` and `SlowDown()`.

☑**Instructions**

1.
Add a `public` `double` `Weight` property with just a getter.

Hint

The format of an automatic property with get only is:

```
bool IsFake
{ get; }
```

2.
Add a constructor to the `Truck` class with two parameters: `double` `speed` and `double` `weight`. It should:

Set the `Speed` property using `speed`

Set the `Weight` property using `weight`

Set a random `LicensePlate` value using `Tools.GenerateLicensePlate()`

Set `Wheels` to `8` if `Weight` is less than 400 and set `Wheels` to `12` otherwise

Hint

To make a random license plate, a utility class is provided for you. Use it in the constructor like so: `Tools.GenerateLicensePlate()`.

Remember that a constructor looks like a method, but there is no return type listed and the method name is the name of its enclosing class:

```
class Forest
{
  public int Area;

  public Forest(int area)
  {
    Area = area;
  }
}
```

3.

Add a `void SpeedUp()` method that increases the `Speed` property by `5`.

Hint

Here's an example method, which increases the `Area` property:

```
public void IncreaseArea(int growth)
{
```

```
    Area = Area + growth;
}
```

4.

Did you get an error? There is no setter for the `Speed` property. Add a private setter to that property.

Hint

Here's an example of a property with get and set methods:

```
bool IsFake
{ get; set; }
```

Without a setter, the `SpeedUp()` method won't be able to access `Speed`, and you'll get an error like this:

```
error CS0200: Property or indexer 'Truck.Speed' cannot be assigned to --
it is read only
```

5.

Add a `void SlowDown()` method that decreases the `Speed` by `5`.