# Design Benefits

* If I need to add a 2-cylinder car I can do so without breaking any hierarchy.
* Using an interface instead of inheritance there is no need for repetitive override methods. Every cylinder implements the iStartEngine interface.
* You could add different types of cars/trucks.
* Code is reusable
* Code is encapsulated.

# Difference between composition and inheritance

* Inheritance creates a hierarchy, composition is more free flowing.
* Composition allows your code to be more reusable.
* Inheritance is used when you have a lot of classes that have common behavior, example(Laptop could be the base class for Apple and HP, then it could go into Macs and ProBook, ect.
* The problem with inheritance is that you will eventually have to type out every single different type of possibility, where as with composition you can make an object COMPOSE OF different classes.
* In c# you can only inherit from one class.