# Creating a Car class

A class is a self-contained piece of software that is able to function on it own.

Apology: UML and some newer languages specify the type after the identifier. C# on the other hand requires that you specify the type and then the identifier.

We will model a car in software.

|  |
| --- |
| **Car**  Class |
| **Fields**  - year : int  - manufacturer : string  - model : string  - isDrivable : bool  - price : double |
| **Methods**  + «constructor» Car(year : int, manufacturer : string, model : string, price : double, isDrivable: bool = true)  + ToString() : string |

## Description of field members:

1. **year**: this private member represents the year value of this object

To restrict access fields are normally decorated with the private modifier.

1. **manufacturer**: this private member represents the manufacturer of this car
2. **model**: this private member represents the model of this car
3. **isDrivable**: this private member indicates if this car be driven
4. **price**: this private member represents the selling price of this car

## Constructor:

1. **public Car(int year, string manufacturer, string model, double price, bool isDrivable = true)**: this is the constructor of this class. It takes five arguments and assigns them to the appropriate fields. You will assume that all the arguments will be sane (i.e. the values will not break your program).

## Description of action member:

1. **public override string ToString()**: this method overrides the same method in the base class. It does not take any argument but it returns a string representing this object. You get to decide how the output will look like. It is expected that all the values be displayed.

In your main method write the code to do the following:

The code to test or exercise your application is called a Test Harness.

1. Create at least four cars and print them. Remember to call the constructor with 4 or 5 parameters.