

# Class Members

Fields and methods inside classes are often referred to as "*Class Members*"

```
public class MyClass
{
    // Class members
    public string color = "red";    // field
    public int maxSpeed = 200;    // field
    public void fullThrottle()    // method
    {
        Console.WriteLine("The car is going as fast as it can!");
    }
}
```

## Fields

In the previous chapter, you learned that variables inside a class are called fields, and that you can access them by creating an object of the class, and by using the dot syntax (.).

The following example will create an object of the Car class, with the name *myObj*. Then we print the value of the fields *color* and *maxSpeed*:

```
public class Car {
    public string color = "red";
    public int maxSpeed = 200;
}

public class Program {
    public static void Main(string[] args)
    {
        Car myObj = new Car();
        Console.WriteLine(myObj.color);
        Console.WriteLine(myObj.maxSpeed);
    }
}
```

# Object Methods

Methods normally belong to a class, and they define how an object of a class behaves.

Just like with fields, you can access methods with the dot syntax. However, note that the method must be public. And remember that we use the name of the method followed by two parentheses () and a semicolon ; to call (execute) the method:

```
public class Car {  
    public void fullThrottle() {  
        Console.WriteLine("The car is going as fast as it can!");  
    }  
}
```

```
public class Program {  
    public static void Main(string[] args)  
    {  
        Car myObj = new Car();  
        myObj.fullThrottle();  
    }  
}
```