C# OOP

OP stands for Object-Oriented Programming.

Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Object-oriented programming has several advantages over procedural programming:

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the C# code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorter development time

Classes and objects are the two main aspects of object-oriented programming. A class is like a “blueprint” - a model to shape objects with. A “Car” can be a class and a “Volvo” can be an object.

In C#, we write classes in the following way:

```
public class Car {

        public string Model {get;set;}

        public string Color {get;set;}

}
```

We can initialize them like this:

```
public class Program {

          public static void Main(string[] args)
          {
                  Car myCar = new Car();
                  myCar.Model = “Nissan”;
           }

}
```

Ofcourse, we can have multiple objects:

```
public class Program {

        public static void Main(string[] args)
        {
                Car myCar1 = new Car();
                Car myCar2 = new Car();
                myCar1.Model = “Honda”;
                myCar2.Model = “Nissan”;
         }

}
```