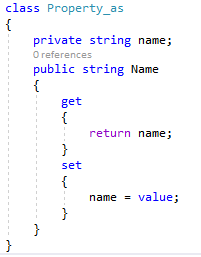
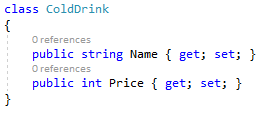
**PROPERTY IN C#**

Property is a member that enables read and write access to the private field using getting and setting property. Get property is used to return property value and set is used to assign new value.



**PROPERTY INITIALIZER**



**CONSTRUCTOR**

Constructor is a method which get invoked automatically when an object of class is created.

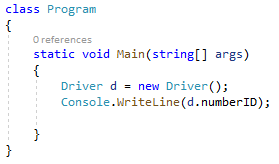
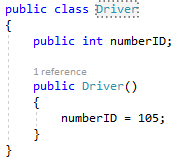
* Constructor must have same name as class.
* Constructor does not have any return type.
* A class can have any number of constructor.

**TYPES OF CONSTRUCTOR**

1. Default Constructor
2. Parameterized Constructor
3. Static Constructor
4. Instance Constructor
5. Private Constructor
6. Copy Constructor

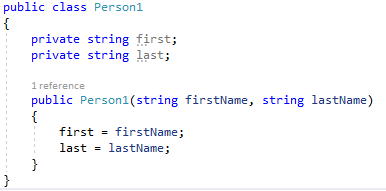
***1. DEFAULT CONSTRUCTOR***

Default constructor is a parameter less constructor. Each class have a default constructor even if it is not declared that instantiates the object and sets member variables to the default values.



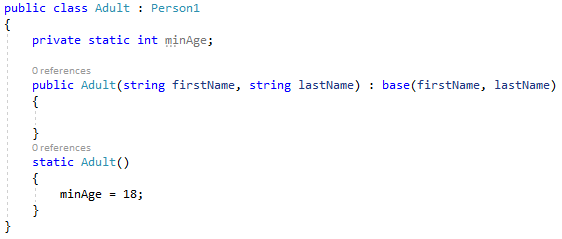
***2. PARAMETERIZED CONSTRUCTOR***

Constructor having same name as class and have parameters in method signature is known as parameterized constructor.



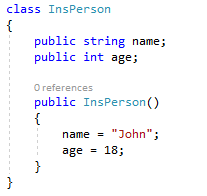
***3. STATIC CONSTRUCTOR***

Static constructor is a parameter less constructor. It get invoked just before the creation of first object of a class. Static constructor cannot be overloaded. Only static field can be accessed within static constructor.



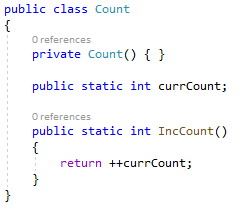
***4. INSTANCE CONSTRUCTOR***

Instance constructors are used to create and initialize any instance member variables the new expression is used to create an object of a class.



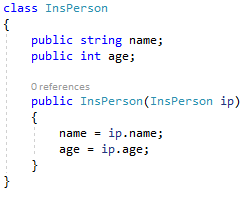
***5. PRIVATE CONSTRUCTOR***

Constructor having private access modifier is known as private constructor. No other class can inherit private constructor method class and define its object. Such kind of class can have only static members.



***6. COPY CONSTRUCTOR***

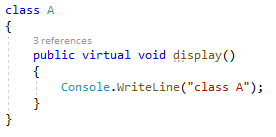
Copy constructor helps in creating an object by copying variables from another object. The purpose of a copy constructor is to initialize a new instance to the values of an existing instance.



**VIRTUAL, OVERRIDE, NEW**

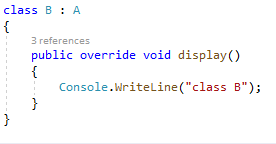
***VIRTUAL***

Virtual keyword is used in parent (base) class with a method which is overridden by child class method.



***OVERRIDE***

Override keyword is used in child class with a method which overrides virtual method of base class.



***NEW***

When child do not wants to override method from base class and want its own method then it uses new keyword with that method.

