Ritik Sharma

- **Property in C#**

Property in C# is just like a data item but here we use 'get()' and 'set()' functions to get the value of property and set the value of property.
It have data type – int, double, string.
We can also have access specifiers on property and also on get() and set() function but on get() and set() access specifier must be narrowed than access specifier on the property itself.
Through access specifiers on get() and set() function we can make a property read-only, write-only or read-write.
Also we get() must return a value. And we can also have our business logic in these functions for easier manipulation of data.
Traditional way to create property:-

```
1   private int _num;
2   public int Num
3       {
4           get
5           {
6               return _num;
7           }
8           set
9           {
10              _num = value;
11          }
12      }
```

Modern Way to create property:-

```
1   public int Num { get; set; }
```

Another Way to create property:-

```
1   private int _num;
2   public int Num {
3       get => _num;
4       set => _num = value;
5       }
```

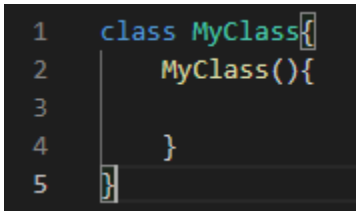
- **Constructor in C#**

Constructor is special method called automatically when object of class is created. And constructors don't have return type.

Constructor types:-

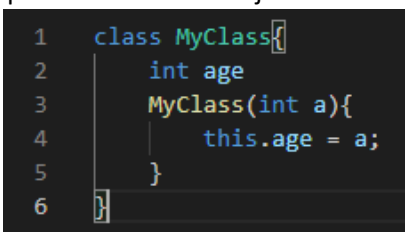
- Default Constructor

  By default, class has a constructor which don't take any parameters but we can also create one. Also used to initialize default value of data members.



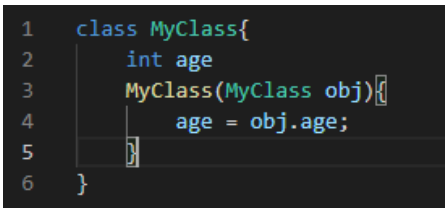
- Parameterized Constructor

  The constructor which take parameters when object is created.



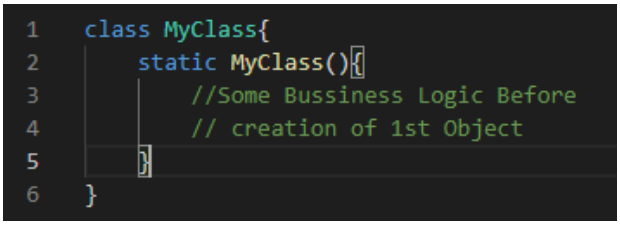
- Copy Constructor

  The constructor which take another instance as a parameter and copy its variables to the newly created instance.



- Static Constructor

  The constructor defined with keyword 'static'. It is called only once when the first object of the class is created. It must be parameter less so we can't overload it. And we can have only 1 static constructor for a class. It gets called first before any type of constructor.



- Private Constructor

  When we define constructor with ‘private’ access specifier, due to which we can’t create object of class containing private constructor as it is private. And also other classes can’t derive it. Usually used where when class has only static members.

```
1   class MyClass{
2       private MyClass(){
3       }
4   }
```

- **Virtual Keyword in C#**

  Virtual keyword is used in base class method so that its child class can override method of child class.

- **Override Keyword in C#**

  Override keyword is used to redefine the same method in base class with virtual keyword so that we can override it in inheritance.

- **New Keyword in Inheritance C#**

  When virtual method of base class is redefined in its child class with new keyword so that child class method has no relationship with base class virtual method as it has its own method with no connection with base class method.

- **Sealed Keyword in C#**

  Sealed keyword is used when we want break inheritance chain, so that class or method with sealed keyword can’t be inherited any further.