Definition

05 July 2019 07:46 PM

Class is nothing but a blue print. Class is reference type and it can have field, property delegate indexer and all
Class is reference type so it will store object in heap.
By default class access modifier is internal…
Other all comes to private ,
Enum and interface is public.
Class - ---

Access modifier -static - return type - Main or class Identifier name ---

Object - > object is basically block of memory that has been allocated to blueprint.
A program can create as many object in same class .
Object are also called instance.
They can be store in array or list.
Object are reference type its store in heap memory.

Example - var obj = new Object();

Only get property can make variable as read only …
Structure - struct - struct are value type-
Struct object can also be created by using the new operator.
But this is not required

Struct object hold the entire copy of struct.

Interface - > interface is type definition similar to class except that is purely a contract between object and user.

Interface is nothing but a collection of method, property declaration.

Interface is public or internal
Interface can declare event and indexer as well.

Abstraction is nothing but showing only the require information and hiding all other information which is unnecessary.
Abstraction can be achieved using private access modifier.

Abstract class is class which is declared by abstract keyword. It can have abstract and non-abstract method. It cannot be instantiate

Implement must be done by derived class.

The purpose of abstract class is to provide a common definition of a class

Class that multiple derived class can share;

Cannot create instance of abstract class.

When a method is declared as a virtual method in a base class then that method can be defined in a base
Class and it is optional for the derived class to override that method.

The override method also provide more the one from for a method
Hence it is also an example of polymorphism.

If there is no virtual, abstract class is there and you have used override keyword in derived class
it will tell the error.

cannot override inherited member 'Base.Test()' because it is not marked virtual, abstract, or override

You can use new keyword to overcome this problem you can use new keyword.
base is keyword in c#.....

Boxing and unboxing ;;;

Value type to reference type converter is called Boxing …
Reference type to value type is called Unboxing.

There is one more keyword which is called - dynamic
Dynamic keyword checked value at Compile time.

It resolved type at run time.

A method can have parameter dynamic type..

Using GetType() you can know the value type in C#.

Delegate default is internal ;;;;
Delegate are type safe similar to function pointer
It is type safe and hold the reference a method.
It can be used with event ….

Default access modifier for class ,struct, Interface, Enum,
Delegate is Internal. Default access modifier for class and
 struct members is private. No access modifier can be
applied to interface members and always interface members are public.

Multithreading - > Multithreading in C# is a process which multiple thread can work simultaneal.
it is a process to archive multitasking
its save time because multiple threaded are being executed at a time.
System. Threading ;

List of commonly classes used in Thread ;
Thread , Mutex, Timer, Monitor, Semaphore, ThreadLocal, Lock, Thread Pool, Volatile

Life cycle --- >
Unstartled, Runnable, Running, Not Running, Dead

Semaphore : Restrict the number of thread can access the a resource.

Mutex : Only one thread to access