

C#

overview of .NET framework

↓
1st released
on
2002

↓
is a software development platform
developed by Microsoft.

↓
the framework was meant to create
applications, which would run primarily
on windows Platform.

↓
can be used to create any kind of
applications

Form based
Web based
Mobile Application.

.NET Architecture

VB - visual Basic

C++

C#

J script

.NET

↓
is a platform for the developers
made up of tools, PL and Libraries.

The framework is a complete
environment that allows developers
to develop, run and deploy application
like →

- console application
- windows forms application
- " presentation Foundation (WPF) application
- web application
- web service
- windows service --- etc.

MSIL - Microsoft intermediate language
এক ধরনের intermediate language code. [যে compiler
আমরা যে code লিখি তার source code কে compile করে
তখন তা হয় intermediate code.]

→ the intermediate code produced by the compiler after
compiling the source code.

This intermediate code is known as MSIL

IL - (intermediate language) is also known as MSIL or CIL
(Common Intermediate language)

JIT - (Just in time Compiler) responsible for converting CIL into
machine code using the common language Runtime
environment.

CLR - (Common Language Runtime). provide an environment to
execute .NET application on target machines.

responsibility of CLR are listed as follows:-

- Automatic memory management
- Garbage collection
- Code access security
- Code verification
- JIT compilation of .NET Code.

Software framework → है एक soft framework पर काज इन एकटा application ऐसी करा, कोत एकटा specific device पर ऊपर base करे

↓
is an abstraction in which software providing generic functionality can be selectively ~~can~~ changed by additional user-written code thus providing application-specific software.

C#

→ was originally developed by a small team by two distinguished microsoft engineers.

Anders hejlsberg
Scott wiltamuth

C# → is a modern, general-purpose, object-oriented programming language developed by MS and approved by ECMA & ISO.

↓
European
Computer
Manufacturers
Association
→ Int
Standard
Org.

C# important point

- C# is case sensitive
- all statement & expression must end with a ";"
- The program execution starts at the main method.
- Unlike JAVA, program file name could be different from the class name.

Example 1st code

using System;] → using keyword, used to include the (system namespace) in the program. A program generally has multiple using statement.

namespace HelloWorldApp] → namespace is a collection of classes.

{

class HelloWorld] → just call declaration. classes contain multiple method. Method defines the behavior of class.

{

Static void Main (String [] args)] → entry point for all C# prog. The main method states what class does when execution.

{

Console.WriteLine ("Hello World");

Console.ReadKey(); // Readline() // ctrl+F5

}

}

It's a keyword which means obj is not required to access static member.

So it saves memory.

memory management help

→ return value

void → return type of the method and it doesn't return any value. In such case (return statement is not required) in C#.

Main → It is the method name. It is the entry point for all C# programs. Whenever we run the C# prog, Main() method is invoked first before any other method. Startup of the program.

String [] args → It is used for command line arguments in C#.

(String type argument)

While running the C#, we can pass value then value receive.

console.WriteLine()

↓
console is the class defined in system namespace.

↓
It is the static method of console class which is used to write the text on the console.
(output text ଏହା ଦ୍ୱାରା)

Class

→ A class is like a blueprint of a specific obj.

example:-

Ferrari is an obj of the luxury car type.

Generally a class declaration contains only keywords
class, followed by an identifier.

↓
unique name

example:- class HelloTAMIM

Components of Class

- i) Modifiers (depends on requirement) (Public/Private)
- ii) keyword class.
- iii) Base/Super class (dep on req)
- iv) Interface (dep on req)
- v) Body { }
- vi) Identifier

obj

It is the basic unit of obj-oriented prog and represents the real life entities.

An obj consists of 3 things

- i) State (Attributes) it is represented by attributes of an obj. Represents the properties of an obj. ex:- Dog
↓
age, height, type
- ii) Behavior
- iii) Identity

It is represented by the method of an obj. It also reflects the response of an obj. ex:-
Barking
Running

It gives a unique name to an obj and enables one obj to interact with another obj. example:- Rocky, Bruno.

constructor special method which get automatically called whenever an obj/instance is created.

Important points

- 1 → must have the same name as the class, which it resides.
- 2 → can not be abstract, final.
- 7 → in a class, you can create only one static constructor.

- 3 → no return type not even void.
- 4 → A class can have any number of constructors
- 5 → Access modifiers can be used in constructor declaration
- 6 → Static constructor can not be parameterized.

Constructor

i) default cons

ii) parameterized cons

iii) copy cons

iv) private "

v) static "

a cons with no parameter

a cons having at least 1 parameter

This cons create obj by copying variable from another obj.

Created with static key word.

Can't be called directly.

When it is executing then the user has no control.

It doesn't take any access modifier.

Cons point 6/7

If a cons is created with private specifier,

it is not possible for the other class to derive from the class.

it is not possible to create an instance of the class.

Main use is to initialize a new instance to the values of an existing instance.

control its access.

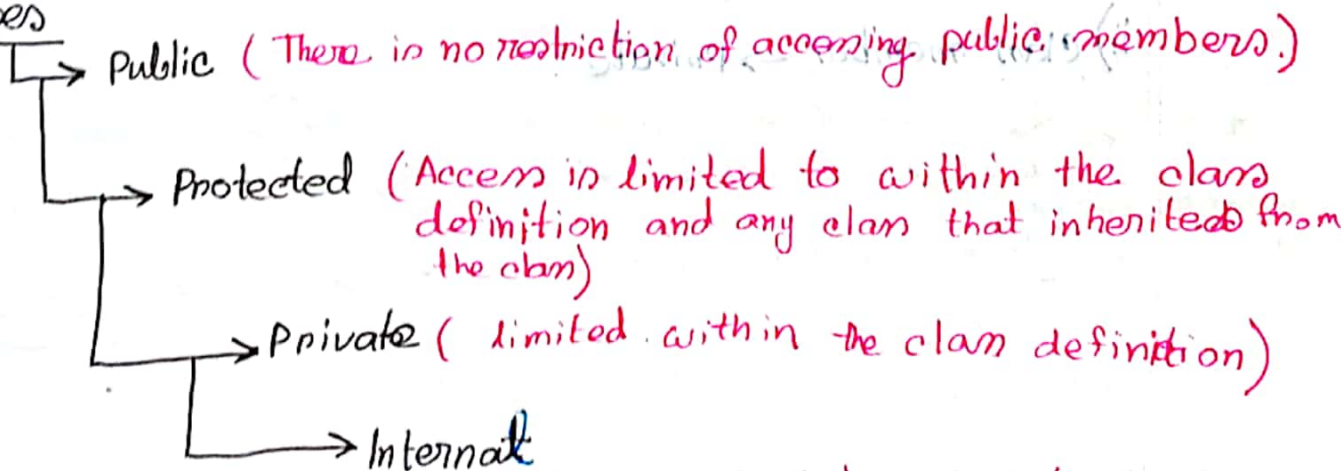
Destructor: special methods inside the class used to destroy instances of the class when they are no longer needed.

- The destructor is called implicitly by .NET framework Garbage collector and therefore programmer has no control as when to invoke the destructor.
- destructor has no return type and exactly same
- It is distinguished apart from constructor because of the tilde symbol (~) prefixed to its name.
- don't except any parameter and modifiers.
- It can not be overloaded or inherited.
- It called when the program exists.
- Unique to its class, not more than one in any class.

Access modifier (Private/Protected)

access modifier are the key word that define the accessibility of a member, class, or datatype in a program.

→ Types



(Access is limited exclusively to classes defined within the current project assembly)

Types of accessibility level

- i) Public
- ii) protected
- iii) private
- iv) internal
- v) protected internal
- vi) private protected

↓

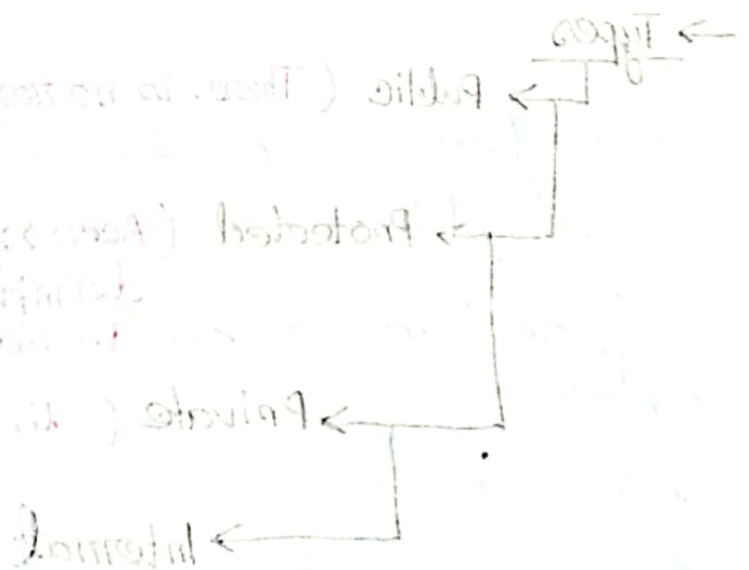
Access is limited to the containing class or types derived from the containing class with the current assembly.

↓

Access is limited to the current project assembly and types derived from the containing class. All members in current project and all members in derived ~~class~~ class can access the variable.

Default access modifier

- i) class → Internal
- ii) class member → Private



Properties:- A property in C# is a member of a class which is used to set and get data from a data field of a class.

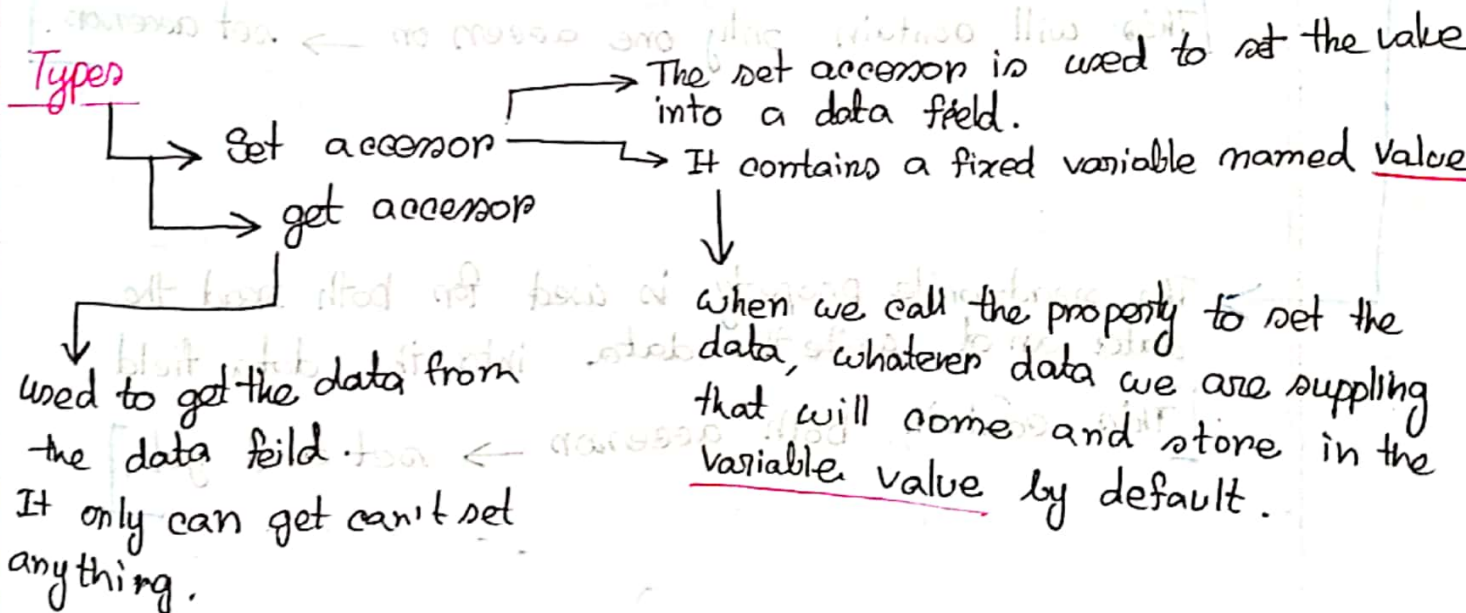
Imp point

The most imp point

- It is never used to store data, it just acts as an interface to transfer the data.
- We use the property as they are the public data member of a class.
- They have special methods called accessors.

What are the Accessors:- Nothing but special methods which are used to set and get the value from the underlying data member.

Types



Different types of properties

→ The C# .Net supports four properties.

1) Read only

The read only is used to read the data from the data field.

[Using this one can not set the data into the field.]
[The property will contain only one accessor → get accessor.]

2) Write only

3) Read-write

4) Auto implemented

If we do not have additional logic while getting the data from the field then we can use of the auto implemented property. It reduces the amount of code we have to write.

→ The write only is used to write the data from the data field.

[Using this one can not read the data from the field.]

[This will contain only one accessor → set accessor.]

→ The read-write property is used for both read the data and write the data into the data field.

[This contain both accessor → set and get.]