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) applicatio
→ web application
→ web service
→ windows service ---- etc.

MSIL – Microsoft intermediate language.
এক ধরনের intermediate ~~languag~~ code. [যখন compiler আমরা যে code লিখি অর্থাৎ 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 intime 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.

environment দেয় যাতে .Net app particular system এ run করতে পারি

responsibility of CLR are listed as follows:

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

Software framework → যে কোন Soft framework এর কাজ হল একটা application তৈরী করা, কোন একটা specific device এর উপর base করে

↓ is

an abstraction in which software providing generic functionality can be selectively ~~cana~~ changed by additional user-writtencode thus providing application-specific software.

C#

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

Ander hejlsberg

Scott wiltamuth

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

ECMA ↓ European Computer Manufacturers Association

ISO ↳ 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~~ Unlike JAVA, program file name could be different from the class name.

Example 1st code

```
using System;
namespace HellowoWorld App
{
    Class HelloWorld
    {
        Static void Main (String [] args)
        {
            Console. WriteLine(" Hello World ");
        } Console . Readkey (); / Readline () // contr+ F5
    }
}
```

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

namespace HellowoWorld App → namespace is a collection of classes.

Class HelloWorld → just call decleration. Classes contain multiple method. Method defines the behavior of cl

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

// contr+ F5 দিয়ে run দিলে তাড়া লাগে না-

Static ← It's a keyword which means obj is not required to access static member. So it saves memory.

memory management এ help করে

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

যদি কিছু return করবে না তখন ব্যবহার হয়.

এই case এ " " প্রয়োজন হয় না.

Main → It is ~~a~~ the method name. It is the entry point for all C# programs.
Whenever we run the C# prog, Main() method is invoke first before anyother method. Startup of the programm.

String [] args → It used for command line arguments in C#
(String type ↓ argument) While runing the C#, we can pass value. then value receive তারা কাজ করে.

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 এ text এর জন্য)

#Class

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

example :-

Ferrari is an obj of the luxury car type.

Generally a class decleration contains only keyword 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.

1) State (Attributes) it is represented by attributes of an obj. Represent 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:- Tom 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 resids.

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 constructor

5 → Access modifiers can be used in constructor diclaration

6 → Static constructor can not be premeterized.

Constructor

i) default cons → a con's with no parameter

ii) parameterized cons → a cons having atleast 1 parameter

iii) copy cons → This cons create obj by copying variable from another obj.

iv) private "

v) Static "

- □ 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 parameters 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 (Privet /Protected)

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

→ Types
- → Public (There is no restriction of accessing public members.)
- → Protected (Access is limited to within the class definition and any class that inheriteds from the class)
- → Private (limited within the class definition)
- → Internal
  (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 current project assembly and types derived from the containting class-
All member in current project and all members in derived ~~cath~~ class can access the Variable

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

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 ~~fie~~ feild of a class.

Imp point

The most imp point

→ It is never used to store data, it just acts as an interface ~~to~~ to transfer the data.

→ We use the property as they. are the public data member of a class

→ ~~They hav~~ It has 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 members.

Types

- → Set accessor
  - → The set accessor is used to set the value into a data feild.
  - → It contains a fixed variable named Value
    - ↓ when we call the property to set the data, whatever data we are suppling that will come and store in the variable value by default.
- → get accessor
  - ↓ used to get the data from the data feild. It only can get can't set anything.

Different types of properties

→ The C# .Net supports four properties :-

1) Read only — The readonly 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 — If we do not have additional logic while setti on getting the data from the field then we can use of the auto implemented property

4) Auto implemeted
→ can use of the auto implemented property
→ 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.]