15/04/2025

Day 42

Memory allocation for class :-

I) Heap memory :-

* when object is created using the new (New) keyword : memory is allocted in Heap.

* The memory is remains allocted until the object is no loger referced and garbage collection remove it.

II) Stack memory :-

* when a method is called local variable (like method Parameters) are Stored in the Stack.

* when the method execution Completes the stack memory is cleared

Variable :-

eg :- → int variable .

int count

↓

default it will be private
so we need give public
to make available in every
class.

every object have reparte
memory allocation .

eg :-

car Tata = New car();

car Toyota = New car();

both object have reparte
memory allocation .

readonly :- (only one time we can change)

public readonly string voucher;

L> can set value at the beginning then it's locked.

L> Eg :-

```
readonly string name;
public person (string name) {
    myName = name;
}
```

* can only set in constructor
* can not change after object is created.

Analog :-

like name we have name in the birth after that we cannot change one time it is fixed.

Const :-

Public Const string Hi ;

↳ Never Changes

↳ Always the same for everyone forever.

eg :-

const double PI = 3.14 ;

↳ value is known at compile time.

↳ can not ever change it

↳ it's automatically static (shared across all objects).

Analogy : Like our data of birth it never changes.

we need use class to access const. eg :- Care . PI .

Static variable (shared by everyone).

Static int totalPeople = 0.

- One copy for all object.
- Belongs to the class, not each object.
- All objects share the same value.

To access use class name.

eg: Car.company = "TCS";

It will show the last value in the company if change.

Volatile variable

(used in multithreading)

volatile bool isRunning;

↳ always get the fresh value from memory;

↳ used when multiple threads access a variable.

↳ prevents the compiler from using old/cached values.

Analogy:-

Like a live scoreboard