

Initialisation List

basics

`int a = 5;` \longrightarrow `int a;`
`a = 5;` | 'a' is created with garbage value then we changed garbage value with '5'.

in case of Constant \rightarrow

`int Const b = a;` \longrightarrow `int Const b;` (X)
`b = a;` (X)
Creating 'b' with a garbage value is not allowed, as it is meaningless to do so.

in case of References \rightarrow

`int i = 5;`
`int &j = i;` \longrightarrow `int &j;` (X)
`j = i;` (X)

\longrightarrow

```
Class Student {  
    Public:  
        int age;  
        Const int RollNumber;  
};
```

dry Run \rightarrow

default Constructor

age = //
Roll.No = //
s1

Student s1;

// \rightarrow garbage

```
int main() {
```

```
    Student s1;
```

```
    s1.age = 20;
```

```
    s1.RollNumber = 101;
```

age = 20;
Roll.No = //
s1

s1.age = 20;

&

\longrightarrow Must give "Error".

\rightarrow but, on Execution we will get Error at "Student s1;" before only, Cause we are not allowed to Create Const with "garbage" value

→ Now If we think of doing, `Const int RollNumber = 101;`, it is useless as all Students will have Same RollNumber then.

→ one feasible solⁿ is, Call the Constructor & in Constructor we need to change it, as Constructor is must while object is created.

So, Lets try this —

Public:

int age;

Const int RollNumber;

Student (int r) {

RollNumber = r;

}

doing this, we get Error —

"Constructor for 'Student' must explicitly initialize the Const member RollNumber"

Lets understand the Error —→

int a;

a = 5;

int a; }
int a = 5;

→ we don't write as
because, 'int a;' Leads to creation of 'a' in memory we need not to create again.

in our Code —→

Const int RollNumber;

& RollNumber = r;

→ this shows, that RollNumber is created in memory before only with some garbage value, & we are trying to change that garbage with 'r'.

→ & to solve now this issue we need to use, Initialisation list —;

=> What initialisation list will do?

→ initialisation list, Jo bhi Constant data members hai Just like "RollNumber" unko Constructor Ke andar Jone se Pahale hi (i.e., at the time of memory allocation only) unki value initialise karke.

→ Lets see how to use it →

Public :

int age;

Const int RollNumber;

```
Student(int r) : RollNumber(r) {  
    }  
    Const int RollNumber = r;
```

int main() {

Student s1(101);

s1.age = 20;

}

& this works fine

In short

NOTE → whenever we have Constant data members, it is must for us to create a Constructor & use initialisation list.

→ we can do this as well,

```
Student() : RollNumber(101) {
```

but this is again useless as, Const int RollNumber = 101,

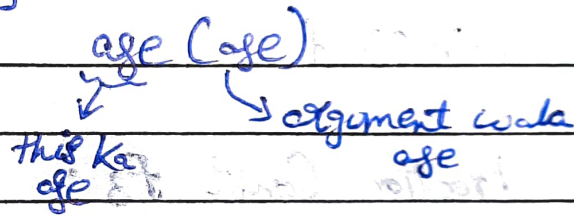
→ we can also initialise non-constant datamembers in initiation list as well

```
Student (int r, int a) : RollNumber(r), age(a) {
```

& can also write as —

```
Student (int r, int age) : RollNumber(r), age(age) {
```

Need not to use this, In fact on using this we might get Error,



→ & we also do this same in case of Reference Variables.

```
Const int RollNumber;
```

```
int & x; // age Reference Variable
```

```
Student (int r, int age) : rollNumber(r), age(age), x(age) {
```

Now, there is a issue,

what we want ———→ int & x = age;

but what happen is ———→ x(argument)

i.e, we need to use this here,

⇒ Student (int r, int age) : RollNumber(r), age(age), x(this → age) {

Now it is fine