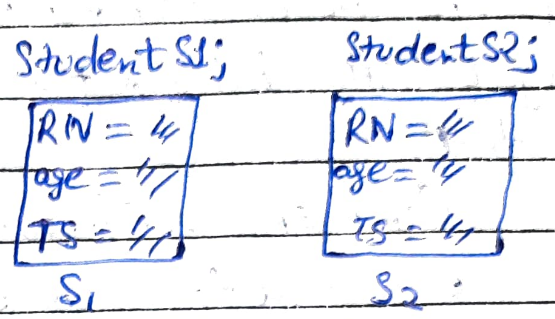
Static Members

```
Class Student {

    Public:
    int RollNumber;
    int age;

    int totalStudents;
};
```



Student S1;

RN = //
age = //
TS = //

S1

Student S2;

RN = //
age = //
TS = //

S2

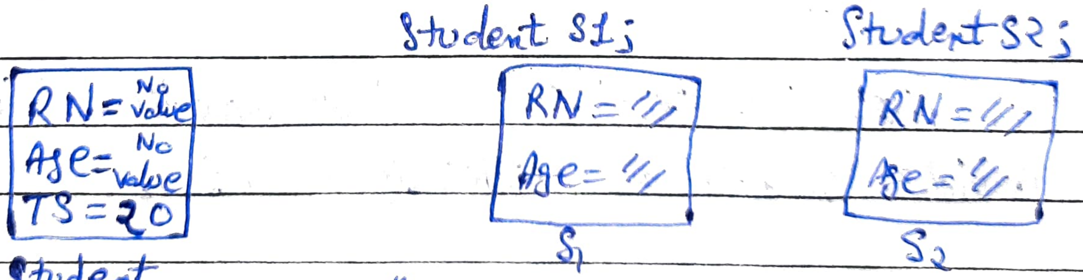
but carrying 'TotalStudents' is useless —
as, TotalStudents must not be a any Single Student "S1", "S2"

→ 'Total Student' is a Property of Class
for that we need to use Static

i.e,

```
Static int totalStudents;
```

now with this code —



RN = No value
Age = No value
TS = 20

Student

Student S1;

RN = //
Age = //

S1

Student S2;

RN = //
Age = //

S2

"Total Student" will not Even got created in S1 & S2.

& we Can Say, RollNumber & age are Non-Static

Now, How to access Static Property?

→ Cout << Student :: totalStudents;

↑ Scope Resolution operator

→ Initialise Value in Static Property?

```
Class Student {
        Public:
        int Ro—;
        int age;
        Static int totalStudents;
};

int Student :: totalStudents = 2;
```

i.e, we have to go out of the class & then initialise Static Property we cannot initialise them inside class.

NOTE:- while initialising we need to write "int" age to initialise value.

→ If we use `Cout << S1.totalStudents << Endl;` then also we get our output of total Students, but that is logically incorrect.

→ we Can also change the Value of "TotalStudents" Easily by using any object too —

```
S1.total Students = 30;            |  output :- 30
Cout << S2.totalStudents << Endl;  |
```

→ ie, change is in class only, there is no other Copy of "TS".

→ Now, the most ImPortant thing that we want is, how to make this totalStudents Such that its value get auto Updated whenever we make a new object (S_1, S_2 ...).

- Lets See —

⟹ The SimPle Solution is use the Constructor, As Castructor is called Every time whenever a new object is created

```
Student() {
    totalStudents++;
}
int Student :: totalStudents = 0;
```

Now, So long we are creating only the data members as static, Lets try to make our functions as Static —

— functions Such as getRollNumber() is Non-Static as for Every object (S_1, S_2 ..) RollNumber is diff..

but, If we make totalStudents as Private, then we need to create getTotalStudents() to get the value, such function we Can make as static, as It is indePendent of objects.

```
Static int gettotalStudents() {
    return totalStudents;
}
```

To access →

```
Cout << Student :: gettotalStudents();
```

→ ImPortant Points to Note about Static functions

- Can only access Static (ProPerties/datamembers/other functions)
- Static functions do not have this → Ⓧ

as, In other functions this holds the "address" of Current object, but as in Case of Static function we dont have that Static datamember in our (S_1, S_2) object. So, as there is no object so there is no this→.