Static Data Member

- **It is *sharable member* to all the objects of same class.**
- Sometimes it is required to share property/attribute between same types of object, Then the member should be defined as static.
- The static data member does not have any direct relation with the object means it is independent from the object.
- It gets storage in special memory area called ***Class Area.***
- Its life is all over the program
- Members can be access anywhere through the class name accordingly access specifier.
- It forms in the memory when class get loaded.

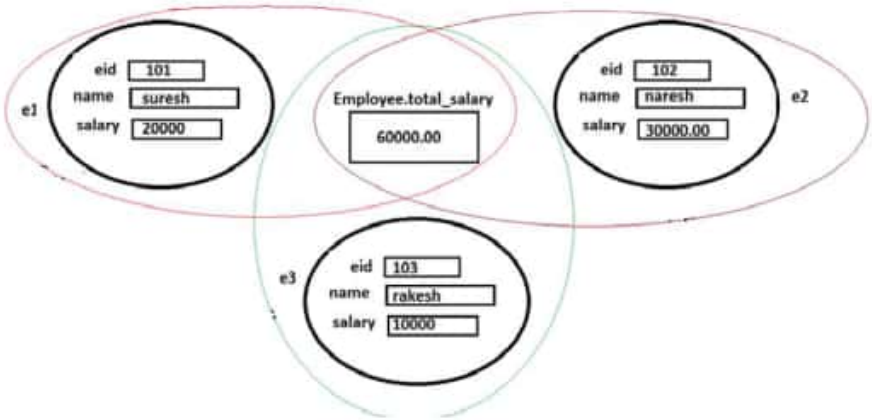
```
#include<iostream>
using namespace std;
class Employee
{
        private:
                int empno;
                char name[20];
                float salary;
                static float total_salary ;
        public:

                void getData()
                {
                        cout << "Enter employee number";
                        cin >> empno;
                        cout <<"Enter employee name";
                        cin >> name;
                        cout << "Enter employee salary";
                        cin >> salary;
                        Employee::total_salary = Employee::total_salary + salary;
                }
                void display()
                {
                        cout << "#1. Employee information";
                        cout << endl << "name\t" << name;
                        cout << endl << "empno\t" <<empno;
                        cout << endl << "salary\t" << salary;

                }
                static void show_total_salary()
                {
                        cout << endl << "total salary\t" << Employee::total_salary ;
                }
};
float Employee::total_salary = 0.0f;
main()
{
                Employee::show_total_salary();
                Employee e1,e2,e3;
                e1.getData();
                e2.getData();
```

```
        e3.getData();

        e1.display();
        e2.display();
        e3.display();
        Employee::show_total_salary();
}
```



- We can also access static members by using object, however it recommended to access it using class name.
  e.g. e1. show_total_salary()

  recommended: Employee:: show_total_salary()

  - such member intialized with 0 by default. however we can initialize with different value such as **float Employee::total_salary = 5.0f;**

- static data member defines twice one inside the class and other outside the class such
- static keyword is not place outside the class static data member definition only inside.
- Object specific member defines as non-static member and class specific define as static membe

Static members function:

- It is also called ***static method***.
- There is no need to form the object (instance) for invoking static method.
- We can define static method by using static keyword as prefix of method.

e.g.

```
class ReviseMyFactorial
{
    public:
    static int factorial(int n)
    {
            int f = 1;
            for ( int i = 1; i<= n; i++ )
            {
                    f = f * i ;
            }
            return f;
    }

};
#include <iostream>
using namespace std;
main()
{
    int n, c;
    cout << endl << "Enter value of n";
    cin >> n;
    c = ReviseMyFactorial::factorial(n);
    cout << "factorial is " << endl << c;

}
```

- Static method we access by using class name with SRO operator as
    - ***int x = ReviseMyFactorial::factorial(n);***
- ***Static method only access static properties/variable.***
- ***Constructor cannot be static.***