Static variables are same as global variable but limited scope .

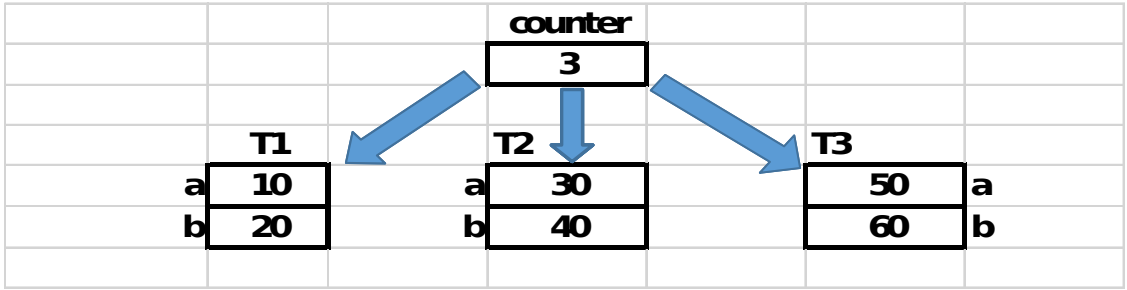
int x=100; static int y=200;

x can be accessible in any file but y will be accessible only in his scope. Static is nothing but shared. When you declared data member as static. It is compulsory to provide global definition for that static data member because static data member always gets memory before creation of object.

```
 #include<iostream>
class StaticDemo
{       int a;
        int b;
        static int counter;
        public:
        StaticDemo(){
                this->_a=0; this->_b=0; counter++;
        }
        void Print() {
        cout<<"_a :: "<<this->_a<<" _b:: "<<this->_b<<endl;
        cout<<" counter" <<this->counter<<endl;
        }
};
```

```
int StaticDemo::counter =0; // global definition for static data member
int main()
{
                StaticDemo T1, T2, T3;
                T3.Print ();
                return 0;
}
```

Size of a object : size of object is size of all non static data members declared in that class. When we declared data member as static at that time instead of getting copy (memory) of static data member to each and every object all the object s share single copy of static data member available in data segment. That's why size of static data member is not consider in to size of object. If you want to shared value of any data member throughout all the objects we should declared data member as static.



Static member function: if we want to call any member function without object name we should declared that member function as static. Static member functions are mainly designed to call using class name only, but we can call static member function by using object name also.
Static member functions can access only static data. Static member functions do not have this pointer.

When we call member function of class by using object implicitly this pointer is pass to that function. When we call member function of class by using class name implicitly this pointer is not pass to that function. Static member functions designed to call by using class name only that's why this pointer is not pass to static members functions of class.

Members that we can call using object of class is called *instance members.*
Members that we can call using class is name these are called as class *members.*

```
#include<iostream>
class Chair
{       private:
                int height;     int width;      static int price;
        public:
                Chair() {
                        this->height=20;
                        this->width=25;
                }
                Chair(int height, int width) {
                         this->height=height ;
                        this->width=width ;
                }
                static void Setprice(int price)  {
                        Chair::price =price;
                }
                void Print()   {
                         cout<<"Height::"<<this->height<<endl;
                        cout<<"Width::"<<this->width<<endl;
                        cout<<"Price::"<<this->price<<endl;
                }
};
```

```
int Chair::price  =125; // global defination for static data member
int main()
{
                Chair ch1, ch2, ch3;
                ch1.Print ();
                ch2.Print ();
                ch3.Print ();

                Chair ::Setprice (200);

                ch1.Print ();
                ch2.Print ();
                ch3.Print ();

                return 0;
}
```

Constant Data Member:

in c language it is not compulsory to initialize constant variable at the time of declaration because we can modify value of const variable using pointer. In c++ it is compulsory to initialize const variable at a time of declaration other wise compile time error will occur.

We cannot initialize data member at the time of declaration. If you want to initialize it in constructor.

```
ConstDemo():a(10), b(20) {

}
 // constructor member initialize list
ConstDemo(int a, int b):a(a), b(b)  {

}
```

when we declare data member as constant it is compulsory to initialize that data member inside constructor initialiser list

Constant member function

we can not declared global function as const. we can declared member function as a const in c++. When we declared member function as const we can not modify the state of as object only with in that const member function.

We should declared member function as a const in which we are not modifying the state of object.

If you want to modify state of the object inside constant member function at that time declared data member as a mutable.

```
#include<iostream>
using namespace std;
class ConstDemo
{
        const int a;   const int b;  int c;
        mutable int d;
        public:
        ConstDemo():a(10),b(20)
        {       c=30;  d=40;
        }
        ConstDemo(int a, int b, int c, int d):a(a), b(b)
        {       this->c=c; this->d=d;
        }
        void Print() const
        {       //this->c=199; //error
                this->d=199;
                cout<<"a::"<<this->a<<endl;
                cout<<"b::"<<this->b<<endl;
                cout<<"c::"<<this->c<<endl;
                cout<<"d::"<<this->d<<endl;
        }
};
int main()
{
    ConstDemo t1;
    t1.Print ();
    ConstDemo t2(1,2,3,4);
    t2.Print ();
    return  0;
}
```