

Static Members

STATIC LOCAL VARIABLE

- declared inside an function
- by default value is 0 not garbage value
- variable lifetime will be throughout the program

```
#include<bits/stdc++.h>
using namespace std;
int func()
{
    static int x; // static local variable
    int y;        // variable
}
```

STATIC MEMBER VARIABLE

- Declared inside the class body
- Also known as class member variable
- They must be defined outside the class
- Static member variable does not belong to any object, but belongs to the whole class.
- There will be only one copy of static member variable for the whole class.
- Any object can use the same copy of class variable
- They can also be used with class name.

```
#include<bits/stdc++.h>
using namespace std;

class Account
{
    private:
        int balance;        // Instance member Variable
        static float roi;    // Static member variable or Class
variable
    public:
        void setBalance(int b)
        {
            balance = b;
        }
}
```

```

        /*
        void setRoi(float r) // Instance Member function
        {
            roi = r;
        }
        */
        static void setRoi(float r) // static member function
        {
            roi = r;
        }
};

// Static member initialisation
float Account::roi = 3.5f; // if we dont give any value, it will be 0
// (:: => scope resolution operator)
int main()
{
    Account a1, a2;
    // Account::roi = 45 // if roi is public
    // BUT what if roi is private ?
    // We make a setter function
    a1.setRoi(4.5f);

    // BUT what if we ant to access the roi without object
    // initialisation ?
    // We should make setRoi function static
    Account::setRoi(4.5f);
}

```

STATIC MEMBER FUNCTION

- They are qualified with the keyword static.
- They are also called class member functions
- They can be invoked with or without object
- They can only access static members of the class