**static variable:**

* static variable has life till the end of the program.
* its value is by default initialized by 0;
* if written inside a function, in case to call function again and again no need to create this variable multipole times as compared n to other local variables.

void fun(){

static int a;//local static variable

int x;//simply local

}

**static member variables:**

* part of the class
* treated as class member.
* not the part of object.
* one copy considered for the whole class.
* must be defined outside of the class.
* if written inside of class then it is called declaration not initialization.
* if defined outside of class then it is initialization.
* any object can take the same value of static variables.
* can be accessed via class name.

class try{

private:

int a;//instance member variable

static float z;//class variable//static member variable

};

float try::z=7.5;//by default 0

int main(){

try t1;//this object have only one variable a

}

**if a static member variable is private it can't be accessed directly into main so we have to make a function to access it.**

main{

try ::z=3.4;//**but not possible here ;otherwise in case of public member we can write this**

* if we want to access a member function then have to make an object first.
* but if want to access without object then that certain public function should be static too.