

C++ Storage Classes

Storage class is used to define the lifetime and visibility of a variable and/or function within a C++ program.

Lifetime refers to the period during which the variable remains active and visibility refers to the module of a program in which the variable is accessible.

There are five types of storage classes, which can be used in a C++ program

1. Automatic
2. Register
3. Static
4. External
5. Mutable

Storage Class	Keyword	Lifetime	Visibility	Initial Value
Automatic	auto	Function Block	Local	Garbage
Register	register	Function Block	Local	Garbage
Mutable	mutable	Class	Local	Garbage
External	extern	Whole Program	Global	Zero
Static	static	Whole Program	Local	Zero



Automatic Storage Class

It is the default storage class for all local variables. The auto keyword is applied to all local variables automatically.

```
{  
auto int y;  
float y = 3.45;
```

```
}
```

The above example defines two variables with a same storage class, auto can only be used within functions.

Register Storage Class

The register variable allocates memory in register than RAM. Its size is same of register size. It has a faster access than other variables.

It is recommended to use register variable only for quick access such as in counter.

Note: We can't get the address of register variable.

```
register int counter=0;
```

Static Storage Class

The static variable is initialized only once and exists till the end of a program. It retains its value between multiple functions call.

The static variable has the default value 0 which is provided by compiler.



```
#include <iostream>
using namespace std;
void func() {
    static int i=0; //static variable
    int j=0; //local variable
    i++;
    j++;
    cout<<"i=" << i<<" and j=" <<j<<endl;
}
int main()
{
    func();
    func();
    func();
}
```

Output:

```
i= 1 and j= 1  
i= 2 and j= 1  
i= 3 and j= 1
```

External Storage Class

The extern variable is visible to all the programs. It is used if two or more files are sharing same variable or function.

```
extern int counter=0;
```

[< prev](#)[next >](#)

Please Share



Join Javatpoint Test Series

Placement	AMCAT	Bank	GATE
Papers	eLitmas	PO/Clerk	NEET
TCS	Java	UPSSSC	CAT
HCL	Python	Government	Railway
Infosys	C	Exams	CTET
IBM	Programming	SSC	IIT JEE
Accenture	Networking	Civil Services	
		SBI	

Learn Latest Tutorials



VBA



SSIS



NGINX