

STORAGE CLASSES

Storage classes decide the behavior/ properties of a variable like

1. **Storage**: In which memory area the variables are stored.
2. **Initial values**: Without initialization, the variable initial (first) values.
3. **Scope**: In which blocks the variables are available i.e. where we can use these variables.
4. **Life time**: Until which time these variables are active (live) in memory.

We have 2 types of Storage classes.

1. Automatic storage class: It contains two specifiers
 - a. auto
 - b. register
2. Static storage class: It contains two specifiers.
 - a. static
 - b. extern

auto variables:

By default all the general / local variables are automatic variables and they created/deleted automatically.

Eg: 1

```
#include<stdio.h>
#include<conio.h>

void main()
{
    auto int a=100; /* int a=100; */
    printf("a = %d\n", a);
    getch();
}
```

Output: a = 100

Eg:2

```
#include<stdio.h>
#include<conio.h>

void fun()
{
    auto int a=10;
```

```
}  
  
void main()  
{  
  
a=20;  
  
}
```

Note: This program gives error because of variable 'a' scope is within function only.

Register variables:

They declared with register keyword.

They stored in **CPU registers**.

Their initial values are garbage.

They are faster than auto variables.

Generally the system is having 4 to 8 registers based on CPU capacity.

When several variables are declared as register variables, the excess variables will become auto variables.

Register variables doesn't have any address. Due to this they are not used in pointer and pointer related operations.

Eg: 1


```
#include<stdio.h>
#include<conio.h>

void main()
{
    register int a = 100;
    printf("a = %d" ,a);
    getch();
}
```

Eg: 2

```
#include<stdio.h>
#include<conio.h>

void main()
{
    register int a = 100;
```

```
printf("a addr is = %u" ,&a);  error
getch();
}
```

Static variables:

The variables that are declared with static keyword are called static variables.

They stored in static area of **data segment**.

Static variables are created/initialized only once at the first time they have been invoked and it is shared by that function several times.

Static variables initial value is **0**.

Their scope is within block / function only.

They are active in memory until total program is executed / closed [life time].

Eg:

```
#include<stdio.h>
#include<conio.h>

void fun()
{
```

```
static int a;  
printf("a = %d\n",++a);  
}  
void main()  
{  
clrscr();  
fun();  
fun();  
fun();  
getch();  
}
```

Output:

a=1

a=2

a=3

Extern variables:

The variables that are declared with extern keyword are called extern variables.

They stored in data segment, which is the public area.

initial value of extern variable is **0**.

extern variables declared inside a function and available to below functions also.

extern variable scope is to all functions declared after the variable.

life time is until total program is executed.

Every extern variable has 2 states.

1.Declared / declaration

Eg: extern int a;

2.Defined / definition

Eg: int a; or int a=10;

The declared part should be conducted within the function, without any initial value.

The defined part should be conducted outside the function, with or without value.

Eg:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void fun1()
```

```
{
extern int  a; /* declared */
printf("a = %d\n",++a);
}

int  a=10; /* defined */
void fun2()
{
printf("a = %d\n",++a);
}

void main()
{
clrscr();
fun1();
fun2();
printf("a = %d",++a);
getch();
}
```


Output:

a = 11

a = 12

a = 13

DIFFERENCES B/W GLOBAL AND EXTERNAL VARIABLES

- Global variables are declared outside the function/top and defined at a time.
- External variables should be declared inside the function and defined outside.
- Global variables doesn't require extern keyword
- Global and external variables available only to the functions declared after the variables.

Storage class identifier	Storage area	Initial values	Scope	Life time
Auto	Stack	Garbage	Within function	Until that function executed
Register	Cpu	Garbage	Within	Until that

	registers		function	function executed
Static	Static area of data segment	0	Within function	Until total program executed
Extern	Data segment	0	Total program	Until total program executed.