Imp
AA

Storage class in C

int a;

A storage class provides some useful information about a variable. This information includes :->

(i) scope
(ii) default initial value
(iii) location
(iv) lifetime

Scope -> It can be local or global
default initial value -> garbage or zero.
location -> RAM or CPU registers
lifetime -> through the program or within a function

A storage class is of 4 types:-

(i) auto storage class
(ii) register " "
(iii) static " "
(iv) extern " "

(4 keywords
⇓
auto, register, static, extern)

Auto storage class :-

* It is the by default storage class of a variable

eg

```
main()
{
 auto int a;
      printf("%d", a);

}
```

scope -> local
default value -> garbage
location -> RAM
lifetime -> within the function

o/p -> garbage value

Register storage class :-

eg
```
main()
{
    register int a;
    printf("%d", a);
}
```

scope → local
default value → garbage
location → CPU Registers
lifetime → within the fxn

o/p → garbage value

(iii) Static storage class :-

eg
```
#include<stdio.h>
void inc();
main()
{
    inc();
    inc();
    inc();
}
void inc()
{
    int a=0;
     a++;
    printf("%d", a);
}
```

a $\boxed{\cancel{0}1}$ a $\boxed{\cancel{0}1}$ $\boxed{\cancel{0}1}$ a

o/p → 1 1 1

```
#include <stdio.h>
void inc();
main()
{
    inc();
    inc();
    inc();
}
void inc()
{
    static int a=0;
    a++;
    printf("%d", a);
}
```

a $\boxed{\cancel{0}\cancel{1}}$ $\cancel{2}$ 3

o/p → 1 2 3

* static variables are initialised only once.
* value will persist b/w various function calls.

Scope → local
default Initial value → zero
location → RAM
lifetime → value persist b/w various function calls

(iv) extern storage class:-

(eg) #include <stdio.h>
extern int a; ~~zero~~
main()
{
printf("%d", a);
}

scope:- global
default value → zero
location:- RAM
Lifetime:- Throughout the program

★ Here, variable a is a part of another file.

Storage class	Scope	default value	Location	Lifetime
auto	local	garbage	RAM	within the fxn
register	local	garbage	CPU register	"
static	local	zero	RAM	value persist b/w function calls
extern	global	zero	RAM	throughout the program

* Difference between local and global variable

Local variable :- * These are the variables which are declared inside a function.

* Their value does not exist outside the function in which they are declared.

Eg

```
#include <stdio.h>
void sum();
main ()
 {
     int a = 20;          // local variable
     printf("%d", a);
 }

void sum()
 {
     int b=10, c=20, sum;    // local variables
       sum = b+c;
 }
```

Global variable :- * These are the variables which are declared outside the function.

* Their value exists throughout the program i.e any of the function can use value of global variable.

Eg

```
#include <stdio.h>
int g = 10;   // global variable
void sum();
main()
{   int a;
    a= g+10;
    printf("%d", a);       // 20
}
void sum()
{
    g=g+20;
    printf("%d", g); // 30
}
```

(g)

```
#include <stdio.h>
int g = 20;          // global
main()
{
      int g = 10,a;      // local
      a = g+5;
      printf("%d", a);        // 15
}
```

AA In case local & global variables are same, preference will be given to the local variable.