

Storage Class

* A storage class defines scope, initial value and lifetime of a variable.

• Memory assigned to a program in a typical architecture can be broken down into 4 segments —

- ① Code // code inside main
- ② Static | global variables // #define will
- ③ Stack // for user defined func
- ④ Heap // DM, could be requested.

* Scope →

- Where will this variable be available?
- Matlab mein ki ander, user defin. func ki ander, ya bahi thi.
 - kaha par upar se hi variable

* Default initial value →

int a;

printf ("%d", a); // to print kya hogi
kyoki maine kuch ini taliye to kya ni.

* lifetime →

- Kab tak up lebth hoga variable

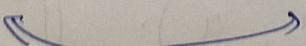
There are 4 types of storage class →

- ① Automatic
- ② External
- ③ Static
- ④ Register

① Automatic Variables | AUTO storage Class

- scope → local to the func. body they are defined in.
- Default value → Garbage value (random)
- lifetime → Till the end of the func. block they are defined in.
- A variable defined without any storage class specification is default an automatic variable.

int harry; and auto int harry;



same

"local variables"

② External Variables | Global Variables | External Storage class

- Scope → global to the program they are defined in.
- Default initial value → 0
- Lifetime → These variables are declared outside any func., they are available throughout the lifetime of the program.
- A global variable can be changed by any func. of the program.
- * int harry; written outside any func. will tell the compiler that harry is a global variable.
- 'extern' keyword →
 - 'extern' keyword is used to inform our C compiler that a given variable is declared somewhere else, using 'extern', compiler will not allocate add'l space for a var.

③ Static Storage class

- Scope → local to the body of func they are defined in
 - default ini. val → 0
 - lifetime → Available throughout the life time of program
 - A static variable remains under existence for use within the func. for the entire program run.
 - Static int harry ; inside any func.

यदि func. bhatam hone ke bad
agr wo func. first call hua to
static variable value retained rakhgi

④ Register Storage class

Scope → Local to the func. they are declared in

Default initial val → Garbage

Lifetime → They are available till the end of the func. block, in which the variable is defined.

* Register values requests the compiler to store variable in the CPU registers instead of storing it in RAM for faster access.

• Size of storage class → lifetime / Time duration , scope / visibility , linkage

* Linkage determines, for a multiple source file program, whether the identifier is known in the current source file or in any source file with proper declaration.

(Bhamu sir notes for storage class)

- An identifier's storage class determines its storage duration, scope and linkage

storage duration → lifetime

scope = visibility

- depending on time period the 4 storage class are split into

① Automatic storage duration

- available till func. is on
 - auto , register

② Static storage dur. →

- available throughout the program run.
 - static , extern

- Global variables and func. names are of 'extern' storage class by default

- array elements aren't automatically initialized to 0.

quick note about array

- double z[100] = {0}

all of the 100 ele.

are init. to 0

```
#define SIZE 5;
creates a constant
'SIZE' and value
stored in it is 5
```

- double w[20] = {5, 3, 1}
 - first 3 have 5, 3, 1
 - rest 0

- bool pass[20] = { true,
 - ↓
 - boolean
 first 3 as T, rest F. true }

A6 3 type of scope note

- Following are the places to declare variables in C →

① Inside the func. or in a block
(local var.)

② In the definition of func. parameters
(formal param.)

③ Outside of all the func. (global.)

eg int main ()
{

int a = 1;

printf a;

{ int b = 2;

③ printf b;

b can't be accessed outside the block

classmate