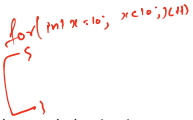
Session 05 B

Thursday, 25 September 2025 9:14 AM

In C programming, **storage classes** define the **scope**, **visibility**, **lifetime**, and **linkage** of variables and functions. Understanding storage classes helps you manage memory usage, variable sharing across files, and variable visibility within your program.
There are **four** main storage classes in C:



🔷 1. auto

- **Default for local variables**
- Used inside functions (i.e., local scope)
- Memory is allocated on the **stack**
- Variable is created when the function is called and destroyed when it exits

Syntax:

```
auto int x = 10;  // Same as just `int x = 10;`
```

Key Points:

- Rarely used explicitly (because it's the default)
- Local scope, no linkage
- Lifetime: during function execution

🔷 2. register

- Suggests the compiler to store the variable in a **CPU register** for faster access
- Used for frequently accessed variables (e.g., loop counters)

Syntax:

```
register int i;
```

Key Points:

- Can't get the address using & operator (since it's in a register)
- Local scope, no linkage
- Lifetime: during function execution

🔷 3. static

- Retains the **value** of a variable **between function calls**
- Can be used in two contexts:
  - Inside a function → retains value between calls
  - Outside a function (global) → makes it **file-scoped**

Syntax:

```
static int count = 0;   // Local static variable
static int x = 100;     // File-level static variable
```

Key Points:

- Lifetime: entire program
- Scope:
  - Local static: function-level
  - Global static: file-level (not visible to other files)
- Linkage: internal (for global static)

🔷 4. extern

- Declares a variable or function **defined in another file**
- Does **not allocate memory**, just references it

Syntax:

```
extern int count;  // Declared elsewhere (another file or later in same file)
```

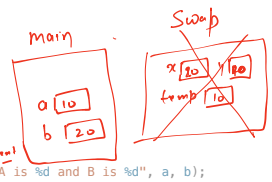
Key Points:

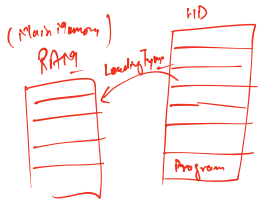
- Used for **cross-file variable/function sharing**
- Linkage: external
- Scope: global
- Lifetime: entire program

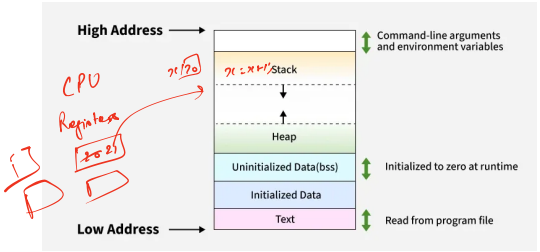
🗒️ Summary Table

Storage Class	Scope	Lifetime	Linkage	Notes
auto	Local	Function call	None	Default for local variables
register	Local	Function call	None	Faster access, no & allowed
static	Local / File	Entire program	Internal	Retains value, not visible outside file
extern	Global	Entire program	External	Declares variable defined elsewhere

```c
#include <stdio.h>
void swap(int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
int main()
{
    int a = 10, b = 20;
    swap(a, b);
    printf("The Value Stored in A is %d and B is %d", a, b);
    return 0;
}
```








```c
#include <stdio.h>
void hello()
{
    static int x = 0;
    printf("The value of x = %d\n", x);
    x++;
}
int main()
{
    hello();
    hello();
    hello();
    hello();
    hello();
    hello();
}
```