

6c - Static Storage Class

Sunday, 9 August 2026 05:04

If local then within `{ }`.

If global then within some file `.c`

eg:

```
fun() {  
    static int a = 5;  
}  
main() {  
    fun();  
}
```

`a` gets loaded into static & `fun` gets initiated in stack.

Once `fun` ends, it is popped from stack but '`a`' persists in static memory.

So it exists but can not be accessed since '`fun`' which accesses it is popped.

```
#include <stdio.h>  
  
void increment() {  
    static int count = 0;
```

- line initialising count only happens once.

```

    count++;
    printf("%d ", count);
}

int main() {
    increment();
    increment();
    increment();
    return 0;
}

```

- increment() starts in stack, finishes and leaves the virtual memory.
- New increment needs it to be allocated memory again.
- But since count is stored in static, its value persists.

Since value is persistent, it keeps incrementing on every function call.