Static Variables in C

In C, static variables are the variables declared using **static** keyword. These variables behaving differently compared to normal or global variables.

Properties of Static Variables

Following are the key properties of static variables in C:

1. Default Value

If you left an uninitialized static variable, then it gets assigned a default value that depends on its type:

- **Static integer variables** default to 0.
- **Character variables** default to \0 (null character).
- **Floating-point variables** default to 0.0.

C

```c
#include <stdio.h>

int main()
{
    static int x;
    printf("%d", x);
    return 0;
}
```

Output

```
0
```

This behaviour makes them predictable compared to normal (non-static) variables, which may hold garbage values if uninitialized.

2. Static Variable Lifetime

Local variables in a function are created when the function is called and destroyed when the function ends. However, if a local variable is declared as static, its lifetime extends beyond the function's scope.

This means that the variable retains its value across multiple function calls. It is initialized only once during the program's execution.

3. Static Global Variables

Normal global variables can be accessed from other files if they are linked appropriately, whereas static global variables have internal linkage. Internal linkage refers to the scope of variables that restricts their visibility to the current file only.