ChatGPT

Global vs Local vs Static Variables in C

This document summarizes the differences, definitions, use cases, and examples of various types of variables in C.

---

1. Global vs Local Variables

Global Variable

- **Declared**: Outside of any function.
- **Scope**: Entire file/program.
- **Lifetime**: Entire program duration.
- **Access**: Accessible from any function in the same file; accessible in other files using `extern`.

Local Variable

- **Declared**: Inside a function or block.
- **Scope**: Only within the declaring function/block.
- **Lifetime**: Exists only during the function call.

Example

```c
#include <stdio.h>

int globalVar = 10;  // Global variable

void function1() {
    int localVar = 5;  // Local variable
    printf("Inside function1: globalVar = %d, localVar = %d\n", globalVar,
localVar);
}

void function2() {
    // localVar is not accessible here
    printf("Inside function2: globalVar = %d\n", globalVar);
}

int main() {
    function1();
    function2();
```

```
    return 0;
}
```

---

2. Static Local vs Static Global Variables

Static Local Variable

- **Declared**: Inside a function with `static` keyword.
- **Scope**: Local to that function.
- **Lifetime**: Entire program duration.
- **Behavior**: Retains its value between function calls.

Static Global Variable

- **Declared**: At the top level with `static` keyword.
- **Scope**: File scope only (not accessible from other files).
- **Lifetime**: Entire program duration.

Example: Static Local Variable

```c
#include <stdio.h>

void counterFunction() {
    static int counter = 0;  // Static local variable
    counter++;
    printf("counter = %d\n", counter);
}

int main() {
    counterFunction();  // counter = 1
    counterFunction();  // counter = 2
    counterFunction();  // counter = 3
    return 0;
}
```

Example: Static Global Variable

```c
#include <stdio.h>

static int hiddenGlobal = 42;  // Static global variable

void show() {
    printf("hiddenGlobal = %d\n", hiddenGlobal);
}
```

```c
int main() {
    show();  // Output: 42
    return 0;
}
```

Summary Table

Feature	Global Variable	Local Variable	Static Local Variable	Static Global Variable
Declared	Outside function	Inside function	Inside function	Outside function
Scope	Whole program	Function only	Function only	File only
Lifetime	Entire program	Function call	Entire program	Entire program
Retains value?	Yes	No	Yes	Yes
Accessible in other files?	Yes (with `extern`)	No	No	No

Use Cases

Variable Type	Use Case Example
Global Variable	Shared configuration or state between many functions.
Local Variable	Temporary calculation or loop variables.
Static Local Variable	Counting function calls, caching results, one-time init logic.
Static Global Variable	Hide global variables from other files (modular design).

Real-life Analogy

- **Global variable**: Like a public park; anyone in the city (file/program) can access it.
- **Local variable**: Like your toothbrush; only used inside your bathroom (function).
- **Static local variable**: Like a fridge in your kitchen; local to you, but keeps food (values) across days (calls).
- **Static global variable**: Like a private garden behind your house; still global in scope for the house but hidden from neighbors (other files).