Programming Fundamentals with C++

Lecture 17 – Functions

Dr. Muhammad Sajjad

RA: Asad Ullah

Overview

- ➢**Local and Global Variables**
- ➢**Local Variables**
- ➢**Global Variables**
- ➢**Static Variables**

Local Variables

- A local variable is declared inside a function or block.
- It can only be accessed within that function or block.
- It does not retain its value once the function exits.
- **Example:**

```cpp
#include <iostream>
using namespace std;

void display() {
    int num = 10;  // Local variable (only exists inside display())
    cout << "Inside display(): " << num << endl;
}

int main() {
    display();
    // cout << num;  // ❌ Error: num is not accessible here
    return 0;
}
```

Key Points

Scope: Available only inside the function/block where declared.

Lifetime: Exists only while the function is executing.

Accessibility: Cannot be accessed outside the function.

Global Variables

- A global variable is declared outside all functions (usually at the top).
- It can be accessed and modified by any function in the program.
- Its value persists throughout the program execution.
- **Example:**

```
#include <iostream>
using namespace std;

int globalVar = 50;  // Global variable

void display() {
    cout << "Inside display(): " << globalVar << endl;
}

int main() {
    cout << "Inside main(): " << globalVar << endl;
    display();  // Accessing the same global variable
    return 0;
}
```

Key Points:

Scope: Accessible in all functions.

Lifetime: Exists throughout program execution.

Modifications: Any function can modify it.

Risk: Global variables can be modified anywhere, making debugging harder.

Static Variables

- A static variable retains its value between function calls.
- It is initialized only once and preserves its last value.
- It is declared using the static keyword inside a function.
- **Example:**

```
#include <iostream>
using namespace std;

void counter() {
    static int count = 0;  // Static variable (initialized only once)
    count++;  // Retains value between function calls
    cout << "Count: " << count << endl;
}
int main() {
    counter();  // Output: Count: 1
    counter();  // Output: Count: 2
    counter();  // Output: Count: 3
    return 0;
}
```

Key Points

Scope: Accessible only inside the function where declared.

Lifetime: Exists throughout program execution but is initialized only once.

Use Case: Useful for counters, caching, and remembering values between calls.

Comparison Table

Variable Type	Scope	Lifetime	Example Use Case
Local	Inside Function/Block	Until function exist	Temporary Calculation
Global	Entire Program	Until Program ends	Configuration settings, Shared counters
Static	Inside Function	Until Program ends	Counters, caching values

Practice Questions

- Write a program to demonstrate the difference between local and global variables.
- Create a function that uses a static variable to count the number of times it is called.
- Modify a global variable inside a function and print its updated value.

Thank You