Global Variables and Scope

In programming , the scope of a variable is defined as the extent of the program code within which the variable can be accessed or declared or worked with. There are mainly two types of variable scopes:

Local Variables

Global Variables

1. **Local Variables** - Variables defined within a function or block are said to be local to those functions.

   Anything between '{' and '}' is said to be inside a block.

   Local variables do not exist outside the block in which they are declared, i.e. they can not be accessed or used outside that block.

   Declaring local variables: Local variables are declared inside a block

```
// CPP program to illustrate
// usage of local variables
#include<iostream>
using namespace std;

void func()
{
    // this variable is local to the
    // function func() and cannot be
    // accessed outside this function
    int age=18;
    cout<<age;
}

int main()
{
    cout<<"Age is: ";
    func();
```

```
    return 0;
}
```

Output

```
Age is: 18
```

2. **Global Variables** - Global Variables can be accessed from any part of the program.

   They are available throughout the lifetime of a program.

   They are declared at the top of the program outside all of the functions or blocks.

   Declaring global variables: Global variables are usually declared outside of all of the functions and blocks, at the top of the program. They can be accessed from any portion of the program.

```
// CPP program to illustrate
// usage of global variables
#include<iostream>
using namespace std;

// global variable
int  global = 5;

// global variable accessed from
// within a function
void display()
{
    cout<<global<<endl;
}

// main function
int main()
{
    display();

    // changing value of global
    // variable from main function
    global = 10;
```

```
    display();
}
```

Output

```
5
10
```

In the program, the variable “global” is declared at the top of the program outside all of the functions so it is a global variable and can be accessed or updated from anywhere in the program.

> 👉 **NOTE - Whenever there is a local variable defined with same name as that of a global variable then the compiler will give precedence to the local variable**