Scope of the Variables

- Scope:
  - Area of program where variable is valid and available to use
  - Mainly two types of variable scopes:
    - Local Variable
    - Global Variable

Local Variables

- Defined within a function or block
- Local variables only exist inside the function or block:
  - Can not be accessed outside that block or function

Local Variables Example

C++

```cpp
// 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;
}

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

    return 0;
}
```

Output:

```
Error: age was not declared in this scope
```

Global variables

- Can be accessed in any part of the program
- Declared at the top of the program outside all functions and blocks

```cpp
#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
```

What will happen if there exists a local variable with the same name as the global variable

- Normally if there are two local variables with same name then compiler generates an error
- But if there is local and global variable with the same name then local variable is given preference
  - Then how to access the global value?

Scope resolution operator

- :: is called scope resolution operator
- It allows us to access the global variable inside a function or block
- Mostly used when there is a local and global variable with the same name

Type Casting and Type conversion

- After Operators