**C++ Variable:**

Lets move on to the types of variables based on the scope.

1. Global variable  
2. Local variable

**Global Variable**

A variable declared outside of any function (**including main as well**) is called global variable. **We can change the value of global inside the function (main also).**

**Program:1**

#include <iostream>

using namespace std;

// This is a global variable

char myVar = 'A';

int main()

{

cout <<"Value of myVar: "<< myVar<<endl;

myVar='Z';

cout <<"Value of myVar: "<< myVar;

return 0;

}

**Output:**

Value of myVar: A

Value of myVar: Z

**Program:2**

#include <iostream>

**using** **namespace** std;

// defining the global variable

**int** a=10;

**int** main()

{

//local variable

**int** a=15;

cout<<"local a: "<<a<<" Global a: "<<::a;

// Re-defining global variable by using ::

**::a=20;**

cout<<"\nlocal a: "<<a<<" Global a: "<<::a;

**return** 0;

}

Output

local a: 15 Global a: 10

local a: 15 Global a: 20

**Program:3**

#include<iostream>

using namespace std;

int global = 5;

void display()

{

    cout<<global<<endl;

}

int main()

{

    display();

    global = 10;

    display();

}

Output:5 10

### Local variable

Local variables are declared inside the braces of any user defined function, main function, loops or any control statements(if, if-else etc) and have their scope limited inside those braces.

#include <iostream>

using namespace std;

char myFuncn() {

// This is a local variable

char myVar = 'A';

}

int main()

{

cout <<"Value of myVar: "<< myVar<<endl;

myVar='Z';

cout <<"Value of myVar: "<< myVar;

return 0;

}

**Output:**  
Compile time error,

## Can global and local variable have same name in C++?

Lets see an example having same name global and local variable.

#include <iostream>

using namespace std;

// This is a global variable

char myVar = 'A';

char myFuncn() {

// This is a local variable

char myVar = 'B';

return myVar;

}

int main()

{

cout <<"Funcn call: "<< myFuncn()<<endl;

cout <<"Value of myVar: "<< myVar<<endl;

myVar='Z';

cout <<"Funcn call: "<< myFuncn()<<endl;

cout <<"Value of myVar: "<< myVar<<endl;

return 0;

}

**Output:**

Funcn call: B

Value of myVar: A

Funcn call: B

Value of myVar: Z