Local variables VS. Global variables

EXAMPLES

```cpp
// 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

```cpp
// CPP program to illustrate
// scope of local variables
// and global variables together
#include<iostream>
using namespace std;

// global variable
int global = 5;

// main function
int main()
{
    // local variable with same
    // name as that of global variable

    int global = 2;
    cout << global << endl;
}
```

output :
2

why ?

BC. if we have 2 variables with the same name
one is local and the other is global ,
the compiler will print the nearest one to the cout

```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;
    cout<<age;
}

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

    return 0;
}
```

Output:

```
Age is: 18
```

```cpp
// C++ program to show that we can access a global
// variable using scope resolution operator :: when
// there is a local variable with same name
#include<iostream>
using namespace std;

// Global x
int x = 0;

int main()
{
  // Local x
  int x = 10;
  cout << "Value of global x is " << ::x;
  cout<< "\nValue of local x is " << x;
  return 0;
}
```

Output:

```
Value of global x is 0
Value of local x is 10
```

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

int sum(int a, int b = 20) {
   int result;
   result = a + b;

   return (result);
}
int main () {
   // local variable declaration:
   int a = 100;
   int b = 200;
   int result;

   // calling a function to add the values.
   result = sum(a, b);
   cout << "Total value is :" << result << endl;

   // calling a function again as follows.
   result = sum(a);
   cout << "Total value is :" << result << endl;

   return 0;
}
```

output:
total value : 300
tatal value :120

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

// Global variable declaration:
int g = 20;

int main () {
   // Local variable declaration:
   int g = 10;

   cout << g;

   return 0;
}
```

output : 10

```cpp
#include <iostream>

using namespace std;

int g = 10;

void func1(){
        g = 20;
        cout << g << endl;
}

int main(){
        func1();
        g = 30;
        cout << g << endl;
        return 0;
}
```

Output

```
20
30
```

```
//  Copyright © 1441 nooralialhomaid. All rights reserved.
//

#include <iostream>
using namespace std;
int variableX=10;
void myFun();
int main()
{
    int variableX=30;
    cout<<variableX<<endl;
    variableX++;
    cout<<variableX<<endl;
    variableX++;
    myFun();
    return 0;
}
void myFun()
{
    variableX++;
    cout<<variableX<<endl;
}
```

output:

30

31

11