



Overwriting the Value of the Global Variable

In this lesson, you will see how to overwrite the value of global variables in a program.

We'll cover the following

- Introduction
 - Example program
 - Explanation
 - function()
 - main()

Introduction#

We know that we can overwrite the value of local variables any number of times in a program. Similarly, we can also overwrite the value of global variables in a function.

Example program#

Run the program below and see the output!

```
1 #include <iostream>
2
3 using namespace std;
4 int global = 3;
5
6 void function () {
7 int function_local = 10;
8 cout << "global = " << global << endl;</pre>
```

```
global = 9;
                                                                    (3)
10
    }
11
    int main() {
12
       int main_local = 20;
13
       global = 5;
       cout << "global = " << global << endl;</pre>
15
       global = 7;
16
       function ();
17
18
       cout << "global = " << global << endl;</pre>
19
20
       return 0;
    }
21
                                                                   \triangleright
                                                                             \leftarrow
                                                                                   X
                                                                               1.1s
Output
 global = 5
 global = 7
 global = 9
```

Explanation#

function()

Line No. 8: The value of global is printed.

Line No. 9: Updates the value of global to **9**.

main()

Line No. 4: Initializes a variable global to **3**.

Line No. 13: In the main function, we overwrite the value of global to 5.





Line No. 14: Prints the updated value of the global.

Line No. 15: We again overwrite the value of global to 7.

Line No. 17: Calls the function in a program.

Line No. 18: Prints the value of global.

Quiz



What is the output of the following code?

```
void function (int local) {
  local = 10;
  cout << "local = " << local << endl;</pre>
}
int main() {
  int local = 20;
  function (local);
  cout << "local = " << local << endl;</pre>
}
```

Your Answer



A) local = 10

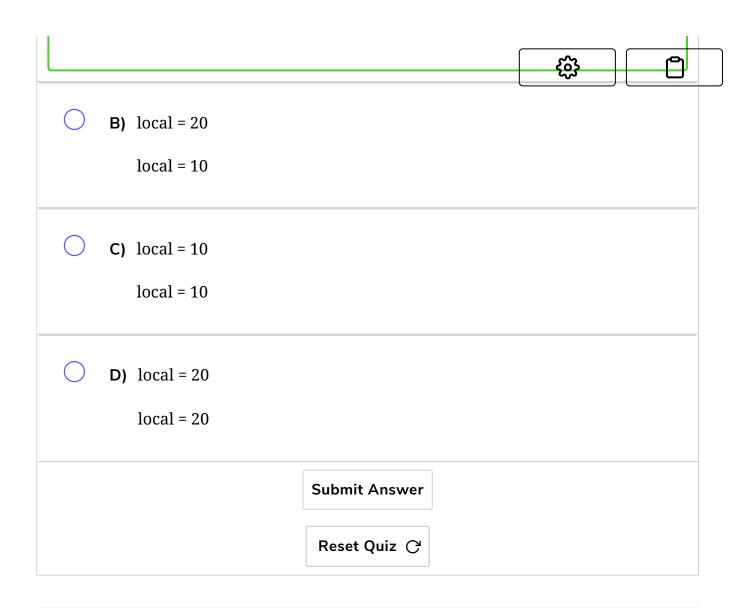


(/learn) local = 20

Explanation

Changing the value of local inside the

function() will not change its value outside the function.



This marks the end of our discussion about functions. Let's solve some challenges related to functions.

