C++ PROGRAMMING

DR. YIHSIANG LIOW (JULY 27, 2025)

Contents

16. Scope

OBJECTIVES

- Understand the scope of a variable
- Understand how C++ searches for a variable
- Understand the importance of using minimal scopes

I' ve already mentioned scopes of variables in several places. In this set of notes, I will collect what you already know about scopes. So much of it is review. I will also add a few new scope rules.

Scope

A variable has a name, a value, and you should know by now, a variable also has a **scope**. This refers to the place in your code where you can refer to that variable.

Here's an old example (very very very old ...):

```
x = 42; // x not declared yet bozo!!!

int x = 0;
std::cout << x << std::endl;
```

In general the scope of a variable is from the **point of declaration** to the **end of the block where it is declared**. During the execution of the program, when the point of execution exits that block, the variable is destroyed. Try this:

```
#include <iostream>

int main()
{
    int x = 0;
    std::cin >> x;

    if (x == 42)
    {
       std::cout << "here we go ... " << std::endl;
        int y = x + 1;
        std::cout << "y: " << y << std::endl;
    }
    std::cout << "x: " << x << std::endl;

    return 0;
```

Scope of y

Of course you should know by now that this won't work (try it):

```
#include <iostream>

int main()
{
     int x = 0;
     std::cin >> x;

     if (x == 42)
```

```
    {
        std::cout << "here we go ..." << std::endl;
        int y = x + 1;
        std::cout << "y: " << y << std::endl;
    }
    std::cout << "y:  " << y << std::endl;

    return 0;
}
```

BAD!!!

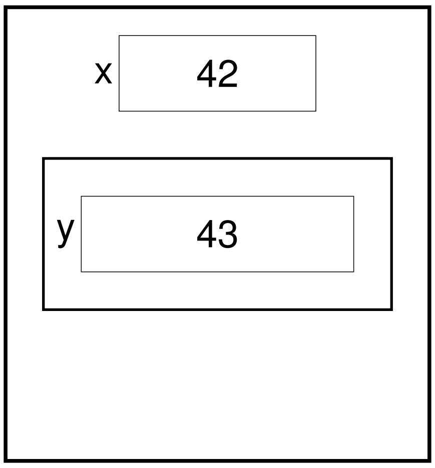
You should visualize the variables in your code as being created in blocks of memory spaces. Suppose the user enters 42 for `x` and we are about to execute the print statement in the body of the `if` statement. The memory looks like this:

```
#include <iostream>

int main()
{
     int x = 0;
     std::cin >> x;

     if (x == 42)
     {
         std::cout << "here we go ..." << std::endl;
         int y = x + 1;
         std::cout << "y: " << y << std::endl;
     }
     std::cout << "y: " << y << std::endl;

     return 0;
}
```



```
#include <iostream>

int main()
{
     int x = 0;
     std::cin >> x;

     if (x == 42)
     {
         std::cout << "here we go ..." <<
         std::endl;
         int y = x + 1;
         std::cout << "y: " << y << std::endl;
     }
     std::cout << "y: " << y << std::endl;

     return 0;
}
```