C++ lambda error

Ask Question

1

1

I saw C++ lambda source.

```
#include <functional>
#include <iostream>

int main()
{
    std::function<int(int)> factorial;

//  factorial = [factorial](int n)->int // runtime error
    factorial = [&factorial](int n)->int // right
    {
        if (n == 1) {
            return 1;
        }
        else {
            return n * factorial(n - 1);
        }
    };

    std::cout << factorial(5) << "\n";
}
```

I don't understand why runtime error occurs.
Thank you for your concerning!

c++ lambda

asked Nov 18 '15 at 9:34

blank_popup
86 1 9

1 Answer

5

✓

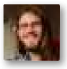
```
factorial = [factorial](int n)->int
```

This version captures `factorial` by value. At the point of the lambda expression `factorial` is empty, so you get a copy of an empty `std::function` and you'll get a `std::bad_function_call` if you try to call it. The `factorial = /*...*/;` is assigning to the original object, which is separate from the copy created for the closure.

```
factorial = [&factorial](int n)->int
```

This version captures `factorial` by reference, i.e. `factorial` inside the lambda names the same object as `factorial` outside. As such, the assignment `factorial = /*...*/;` affects the `factorial` variable inside the lambda, so you call a valid function.

answered Nov 18 '15 at 9:40



TartanLlama
54.4k 10 116 166

Thank you for your help. I would like to answer as you. – blank_popup Nov 19 '15 at 0:28