


C++ 返回值return为引用时报错为局部变量local variable

 C++ 专栏收录该内容

2 订阅 25 篇文章

订阅专栏

我们来研究一下如下函数：

```
1  int &func1() {
2      int i;
3      i = 1;
4      return i;
5  }
```

错误1： 这里func1里的i是局部变量，退出函数后就消失了，你引用了一个消失的变量当然会出问题

```
#include <iostream>

using namespace std;

int &func1() {
    int i; // 这里的i是局部变量，退出函数后就消失了，你引用了一个消失的变量当然会出问题
    i = 1;
    return i;
}

int main() {
    cout << func1();
    return 0;
}
```

warning: reference to stack memory associated with local variable 'i' returned

错误2： 参数传引用时，必须接受一个实际的变量，不然也是在引用一个临时的东西

```
#include <iostream>

using namespace std;

int& func1(int &i) {
    i = 1;
    return i;
}

int main() {
    int a;
    cout << func1(3) << endl;
    return 0;
}
```

error: no matching function for call to 'func1'

正确做法

定义一个参数，接受一个main代码中的变量，并引用它

代码：

```
1  #include <iostream>
2
3  using namespace std;
4
5  int &func1(int &i) {
6      i = 1;
7      return i;
8  }
9
10 int main() {
11     int a;
12     cout << func1(a) << endl;
13
14     func1(a) = 77; // func1(a)的返回值引用了a所以等于a = 77
15     cout << a << endl;
16     func1(a) ++;
17     cout << a << endl;
18     return 0;
19 }
```

输出：

```
1  1
2  77
3  2
```