

C++面试题补充

1、谈谈const 类型的引用？

如果引用的参数是const，则编译器在某些情况下会生成临时变量，比如下面的两种情况

- 1、实参类型不正确，到那时可以转换为正确类型
- 2、实参类型正确，但不是左值

直接看下面的例子：

```
1 void g(string &a, const string &b)
2 {
3     cout << "a address :" << &a << endl;
4     cout << "b address :" << &b << endl;
5
6     // return a;
7 }
8
9 int main()
10 {
11     string input = "I love you";
12     cout << "input address: " << &input << endl;
13     const char *b = "***";
14     cout << "b address: " << &b << endl;
15
16     g(input, b);
17
18     getchar();
19     return 0;
20 }
```

```
input address: 004FFDD8
b address: 004FFDCC
a address :004FFDD8
b address :004FFCE8
```

我们可以看到g的参数一个是引用，一个是const引用；

从截图可以分析出，可以看到将字符串常量传给g函数的const string&参数打印出的地址与传入的地址不同

说明函数内部自动生成了一个临时变量将字符串常量自动转化为string类型。（会临时生成一个变量）

再看另一个例子：

```
1 string f();
2 string g(string & str);
3
4 g(f());    // 出错
5 g("abc"); // 出错
```

上面的程序在4,5行出错。

第四行：将一个临时变量作为实参传递给参数时，是需要将参数类型设为const的。

第五行：当引用参数与实参不匹配，但可以通过转化为引用参数类型时，要将引用声明为const。

总结：

当C++中当函数参数为引用时

- 1、如果传递的实参与函数参数类型不匹配，那么只有将参数类型定义为const，那函数将会产生一个临时变量，临时变量自动转化为函数参数类型。否则将报错。
- 2、如果传递的实参是一个临时变量，那么就要将函数定义为const类型，否则也会出错。

