17 REFERENCES in C++

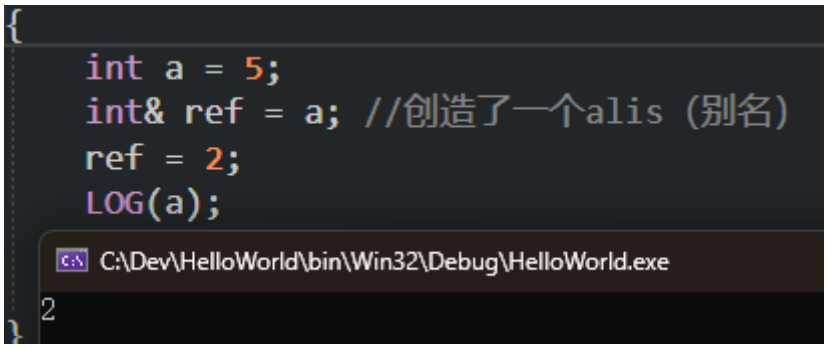
1. 和pointer的关系

事实上，*reference*(引用)只是指针的一个拓展，只是基于指针的一种*syntax sugar*(语法糖)，来使代码更容易读写而已。
顾名思义，引用就是指对现有变量引用的一种方式。
没有*reference*能做而*pointer*不能做的事

和指针不同（先创建一个指针变量，然后赋值nullptr等），*reference*必须引用一个已存在的变量，引用本身并不是一个新的变量，并不真正占用内存。

2. 定义引用

通过**类型&**来实现，此处&是类型的一部分，因此并不一定有&就一定是取地址或一定是引用，具体情况要看*context*



在任何情况下，ref就是a，我们只是给a创建了一个*alias*(别名)，让代码更好写一些。

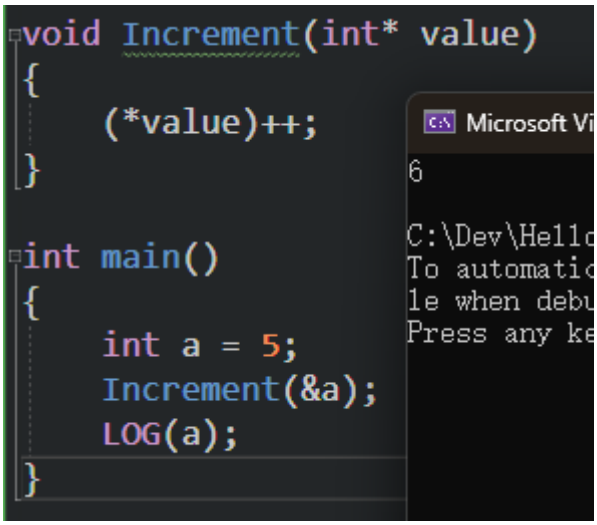
3. 举例

```C++
void Increment(int value)
{
    // int value = 5;
    value++;
}

int main()
{
    int a =5;
    Increment(a);
    LOG(a); // 5
}
```

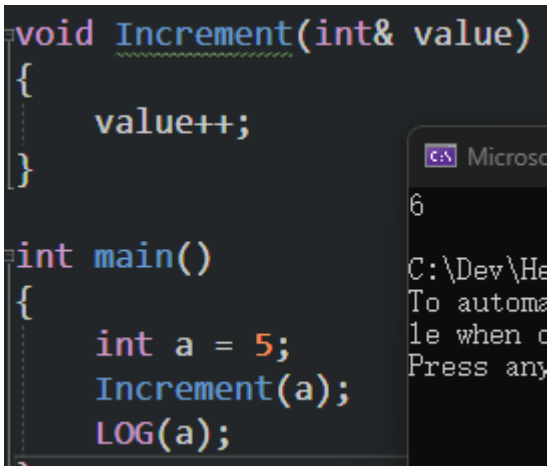
这里我们用的是*passing by value*(传值调用)，并不是**`int& 或者 int*`**，因此会发生的是拷贝一个新的变量=5,并不会改变a。

我们需要的是*passing by reference*(引用传递)，来让这个变量递增



如上图，先解引用，再递增那个地址的数值。但是看着很麻烦，所以可以用如下方式：

用reference的方式



代码更清楚简洁了，增强代码可读性

4. 其他

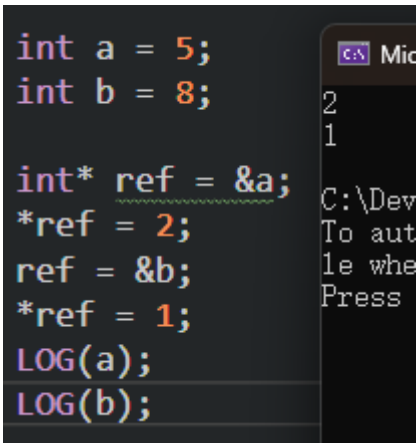
一旦你声明了一个引用，你就不能更改它所引用的对象

```C++
int a = 5;
int b = 8;

int& ref = a;
ref = b;  // a=8,b=8
```

除此之外，一旦声明必须立即赋值，因为他是变量的引用而不是一个真的变量

如果想修改引用，则可以通过指针的方式（指针可以改变指向的对象，而引用不行）


int a = 5;
int b = 8;
int* ref = &a;
*ref = 2;
ref = &b;
*ref = 1;
LOG(a);
LOG(b);
2
1
C:\Dev
To aut
le whe
Press