# cin,cout,以及其返回值

今天在看c++primer的时候,读到其中这样一段话:

When we use an istream as a condition, the effect is to test the state of the stream. If the stream is validthat is, if it is still possible to read another input then the test succeeds. An istream becomes invalid when we hit[end-of-file](http://writeblog.csdn.net/ch01lev1sec8.html#gloss01_16) or encounter an invalid input, such as reading a value that is not an integer. An istream that is in an invalid state will cause the condition to fail.

其对应代码是:while (std::cin >> value)  
开始就觉得这样有点不对劲,但是也不知道是哪里不对劲.仔细一想,原来是觉得这个cin的位置十分诡异...

原来我们最常用的就是直接输入或者直接输出.比如cin>>value;或者cout<<value;现在把这个放到while里面进行判断.

难道cin与cout的返回值是bool型的?好像也说不过去.

GOOGLE之~~

分下面几点来说明:

1.cin和cout是iostream类的2个对象,而对象是无所谓返回值的.有返回值的是<<还有>>这2个操作符. 由于我们知道,操作符其实也就是函数(在操作符重载的时候可以清晰的认识到).而>>操作符返回的是它的左操作数(left- operand).对于cin>>value;返回左操作数就是操作的流的引用,也就是istream&.

2.但是好像还是不对,因为while里面判断的是bool值,难道还能判断istream&吗?

打开<ISTREAM>头文件，找到类模板basic\_istream的定义，摘出这么两个语来：

 typedef basic\_istream<\_E, \_Tr> \_Myt;

           \_Myt& operator>>(......) ......

    这说明cin >>的返回值类型就是basic\_istream&，可是放到while()中情况又该是怎样的。while()中要求是布尔表达 式，难不成basic\_istream&类型可以转换成bool类型？继续查看头文件，发现所有的operator重载函数都是<< 和>>，没有找到用于类型转换的操作那就只好追溯到父类basic\_ios了。

    打开头文件<IOS.H>，找到ios的定义，其中有这么一条语句，类型转换函数的定义：

 operator void \*() const { if(state&(badbit|failbit) ) return 0; return (void this; )

有这个函数的定义之后，编译器会在需要的情况下将ios类型自动转换为 void\*类型。因此，在表达式while (cin >> m >> n)中，括号中的表达式为了匹配bool类型将自动转换为void\*类型。

如果读入时发生错误返回0，否则返回cin的地址。