# Lab3

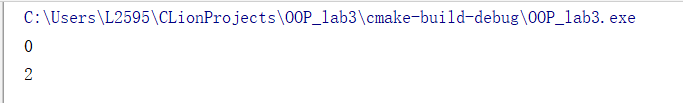
**任务1**

内部连接：如果一个名称对编译单元(.cpp)来说是局部的，在链接的时候其他的编译单元无法链接到它且不会与其它编译单元(.cpp)中的同样的名称相冲突。**例如static函数，inline函数等**（注 : 用static修饰的函数，本限定在本源码文件中，不能被本源码文件以外的代码文件调用。而普通的函数，默认是extern的，也就是说，可以被其它代码文件调用该函数。）

外部连接：如果一个名称对编译单元(.cpp)来说不是局部的，而在链接的时候其他的编译单元可以访问它，也就是说它可以和别的编译单元交互。 例如变量就是外部链接， 全局变量。

**任务1.1**

运行结果如下：



原理分析：

#include **<iostream>** #include **"header.h"** #include **"file1.h"** #include **"file2.h"  
 int** variable2;  
  
 **int** main() {  
 function1();  
 function2();  
  
 std::cout << variable << std::endl;  
 std::cout << variable2 << std::endl;  
 **return** 0;  
 }

如上为main.cpp中内容，可以发现，依次include iostream，以及自定义的头文件header.h、file1.h、file2.h。

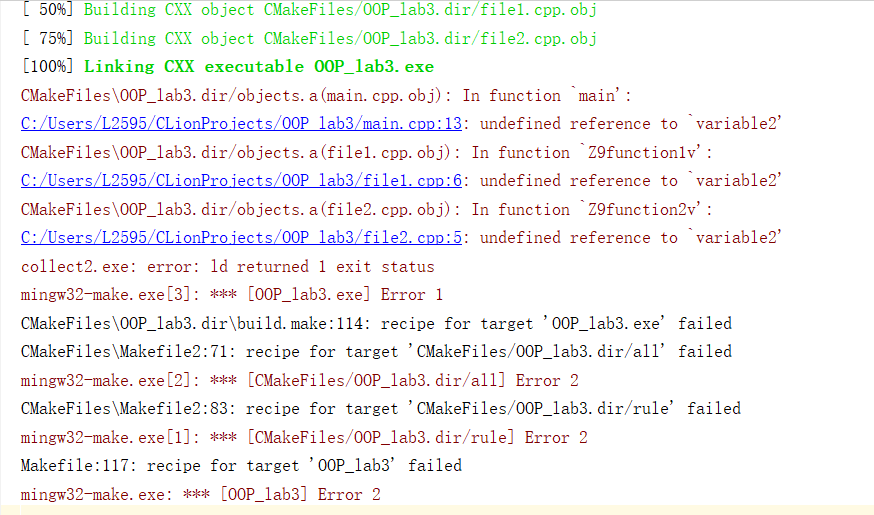
header.h中有static的int变量variable，故其值0。由于该变量为静态的，且是全局的，因此function1和function2并不会改变variable的真正的值，而只是在函数内部暂时改变。而variable2在其中声明，在main.cpp中定义，随着function1和function2函数执行，改成了2。

**任务1.2**



说明：extern在其他文件未声明定义的情况下只起声明作用，因此variable2还未定义便直接赋值0，于是报错。

**任务1.3**



说明：程序报错主要是由于variable2变量没有定义，在函数中赋值和控制台输出时都是不可行的。