## 实验7 Exception Handling

## 学号: 09017423 姓名：杨彬

### 实验目的

Objectives

1. What exceptions are and when to use them.

2. To use try, catch and throw to detect, handle and indicate exceptions, respectively.

3. To process uncaught and unexpected exceptions.

4. To declare new exception classes.

5. How stack unwinding enables exceptions not caught in one scope to be caught in another scope.

6. To handle new failures.

7. To use auto\_ptr to prevent memory leaks.

8. To understand the standard exception hierarchy.

### 实验内容

**Exp1:（习题16.25，异常处理的逻辑流程）**

1． Description of the Problem

（英文）Suppose a program throws an exception and the appropriate exception handler begins executing. Now suppose that the exception handler itself throws the same type of exception. Does this create infinite recursion? Write a program to check your observation.

（中文）假设一个程序抛出一个异常，而一个特定的异常处理程序将开始执行。现在假设一个异常处理程序本身又抛出了一个相同的异常，这会形成无限循环吗？编写一个程序来证明你的观点。

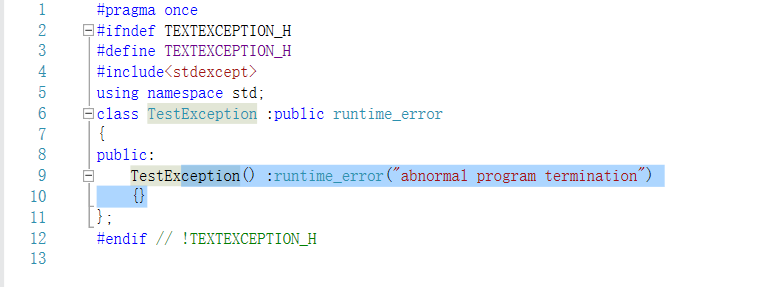
**Exp2:**（习题16.30 构造函数、析构函数和异常处理）

1. Description of the Problem

（英文）Write a program illustrating that member object destructors are called for only those member objects that were constructed before an exception occurred.

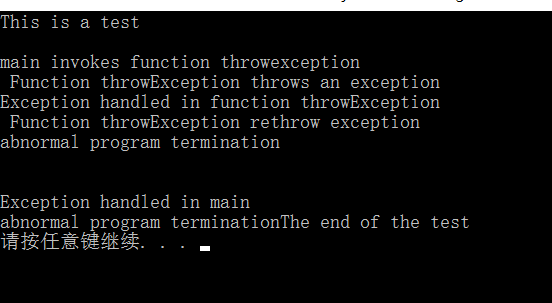
（中文）编写一个程序，证明只有在异常抛出之前创建的成员对象的析构函数才会被调用

### 实验代码及结果：

**Exp1:** 

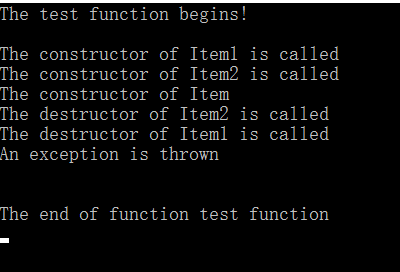






**Exp2:** 





### 遇到的问题及解决过程

**Exp1:**nothing

**Exp2:**

### 实验体会

**Exp1:**通过这个实验我理解了一些异常在传递过程中的特点，异常在传递的时候，如果在里层的try catch 语句块中，try语句throw出了一个异常但是catch语句无法接住，那么这个异常将会被交由上一级来处理，如果catch语句中又抛出了一个异常，那么同样也是由上一级处理。如果每一层都不能解决，那么最终将会传递到main函数处，如果main函数没有catch语句或者无法捕捉异常那么程序就会报错。同时catch 语句throw语句没有指定抛出什么异常的时候，默认抛出相同的异常。

**Exp2:**这个实验证明了一点，在try catch语句中如果碰到了throw 语句那么try中接下来的语句将不会被执行，在本实验中抛出错误之后，类的构造函数无法自己处理，所以让上级处理，Item3的构造函数没有完全执完，没有被构造自然也不会被析构了。同时我们还注意到了类的作用域的问题，这三个类的作用域是try语句，抛出异常之后离开了try 语句，这时候类自然被析构掉了，所以是先打印析构语句，再打印异常语句。可以这么简单理解：两个变量的作用域就是离它最近的两个大括号。（要时刻记住）