**析构函数与构造函数的调用顺序**

对象是由“底层向上”开始构造的，当建立一个对象时，首先调用基类的构造函数，然后调用下一个派生类的构造函数，依次类推，直至到达派生类次数最多的类的构造函数为止。因为，构造函数一开始构造时，总是要调用基类的构造函数，然后才开始执行其构造函数体，调用直接基类构造函数时，如果无专门说明，就调用直接基类的默认构造函数。在对象调用析构函数时，其顺序正好相反。

实例代码：

#include <iostream>

using namespace std;

//基类

class Grandfather

{

public:

Grandfather(){cout<<"constructor - Grandfather! "<<endl;}

~Grandfather(){cout<<"deconstructor - Grandfather! "<<endl;}

};

//派生类

class Father : public Grandfather

{

public:

Father(){cout<<"constructor - Father! "<<endl;}

~Father(){cout<<"deconstructor - Father! "<<endl;}

};

class Son : public Father//继承Father类，三层结构

{

public:

Son(){cout<<"constructor - Son! "<<endl;}

~Son(){cout<<"deconstructor - Son! "<<endl;}

};

//实验主程序

void main()

{

Son son;

}

结果截图：