**Stack Unwinding**

源代码：

# include <iostream>

using namespace std;

class Test {

public:

Test(int a=0) {

x = a;

cout << "Construction"<<x<< endl;

}

~Test()

{

cout << "Destruction"<<x<< endl;

}

private:

int x;

};

void function3() throw (runtime\_error)

{

cout << "In fun3\n";

Test t(3);

throw runtime\_error("runtime\_error in fun3");

cout << "Reach here? fun3\n";

}

void function2() throw (runtime\_error)

{

Test t(2);

cout << "fun3 is callled inside fun2\n";

function3();

cout << "Reach here? fun2\n";

}

void function1() throw (runtime\_error)

{

Test t(1);

cout << "fun2 is called inside fun1\n";

function2();

cout << "Reach here? fun1\n";

}

int main()

{

try {

cout << "fun1 is called inside main\n";

function1();

cout << "Reach here? fun main\n";

}

catch (runtime\_error &error) {

cout << "Exception occurred:" << error.what() << endl;

cout << "Exception handled in main\n";

}

return 0;

}

运行结果：

