

第八章习题

8-2:

信息。

2. 将例 8.3 程序改为下面的程序,请分析执行过程,写出运行结果,并指出由于异常处理而调用了哪些析构函数。

```
#include<iostream>
#include<string>
using namespace std;
class Student
{public:
    Student(int n,string nam)
    class Student
        {cout<<"constructor-"<<n<<endl;
          num=n;name=nam;}
    ~Student() {cout<<"destructor-"<<num<<endl;}
    void get_data();
private:
    int num;
    string name;
};
void Student::get_data()
```

```
{if(num==0) throw num;
else cout<<num<<" "<<name<<endl;
cout<<"in get_data()"<<endl;
}

void fun()
{Student stud1(1101,"Tan");
stud1.get_data();
try
{Student stud2(0,"Li");
stud2.get_data();
}
catch(int n)
{cout<<"num="<<n<<"error!"<<endl;}
}

int main()
{cout<<"main begin"<<endl;
cout<<"call fun()"<<endl;
fun();
cout<<"main end"<<endl;
return 0;
}
```

上机代码与运行结果：

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class Student {
7  public:
8      Student(int n, string nam) {
9          cout << "constructor-" << n << endl;
10         num = n; name = nam;
11     }
12     ~Student() {
13         cout << "destructor-" << num << endl;
14     }
15     void get_data();
16 private:
17     int num;
18     string name;
19 };
20
21
22 void Student::get_data() {
23     if (num == 0) throw num;
24     else cout << num << " " << name << endl;
25     cout << "in get_data" << endl;
26 }
```

```

28 void fun() {
29     Student stud1(1101, "Tan");
30     stud1.get_data();
31     try {
32         Student stud2(0, "Li");
33         stud2.get_data();
34     }
35     catch (int n) {
36         cout << "num = " << n << "error!" << endl;
37     }
38 }
39
40 int main() {
41     cout << "main begin" << endl;
42     cout << "call fun()" << endl;
43     fun();
44     cout << "main end" << endl;
45     return 0;
46 }

```

```

destructor-0
num = 0, error!
destructor-1101
main end

```

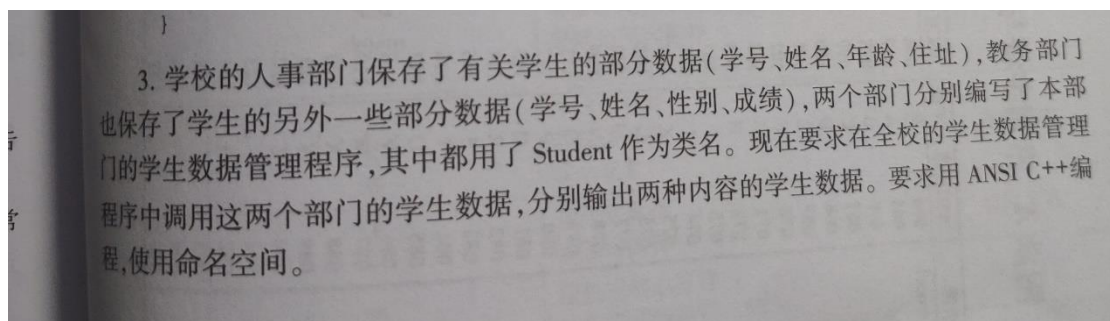
分析：

代码首先执行 main 函数，输出"main begin"，接着输出"call fun()"，表示要调用 fun 函数，流程转到 fun 函数去执行。在 fun 函数中先定义对象 stud1，此时调用 stud1 的构造函数，输出"constructor-1101"，并将 1101 和"Tan"，分别赋给 num 和 name，然后调用 stud1 的 get_data 函数，流程转到 stud1.get_data 函数中去执行。由于 stud1 的 num=1101，不等于 0，因此输出"1101 Tan"，接着执行 get_data 函数中的最后一行 cout 语句，输出"in get_data()"，表示当前流程仍在 get_data 函数中，执行完 stu1.get_data()函数后，流程转回到 fun 函数。

接着执行 fun 函数中 try 块中的语句，定义对象 stud2，此时调用 stud2 的构造函数，输出"constructor-0"，并将 0 和"Li"分别赋给 num 和 name。然后调用 stud2 的 get_data 函数，由于 stud2 中的 num 等于 0，因此执行 throw 语句，抛出 int 型变量 num，此时不会输出 num 和 name 的值，也不执行 get_data 函数中最后一行的 cout 语句，流程转到调用 get_data 函数的 fun 函数去处理。

由于在 fun 函数中有 catch 处理器，catch 处理器捕获异常信息 num，并将 num 的值赋给了变量 n，此时流程脱离 try 块，系统开始析构工作，对于从相应的 try 块开始到 throw 语句抛出异常信息这段过程中已构造而为析构的局部对象（本程序中为 stud2）进行析构，输出"destructor-0"，然后再执行 catch 处理块中的语句，输出"num=0,error!"。fun 函数已执行完毕，在流程转回 main 函数之前先调用 stud1 的析构函数，输出"destrutor-1101"，最后执行 main 函数中最后一行 cout 语句，输出"main end"。

8-3:



上机代码与运行结果:

1_Programing Language > Cpp > Practice > HomeWork > book > 221228_3 > C++ main.cpp > ...

```
1 // main.cpp
2 #include "xt8_3_h1.h"
3 #include "xt8_3_h2.h"
4 #include <iostream>
5
6 using namespace std;
7 using namespace student1;
8
9 int main() {
10     Student stud1(1001, "Wang", 18, "123 Beijing Road, Shanghai");
11     stud1.show_data();
12     student2::Student stud2(1102, "Li", 'f', 89.5);
13     stud2.show_data();
14     return 0;
15 }
```

1_Programing Language > Cpp > Practice > HomeWork > book > 221228_3 > xt8_3_h1.h > ...

```
1 // File: xt8_3_h1.h
2 #include <iostream>
3 #include <string>
4
5 namespace student1 {
6     class Student {
7     public:
8         Student(int n, std::string nam, int a, std::string addr) {
9             num = n; name = nam; age = a; address = addr;
10         }
11         void show_data();
12     private:
13         int num;
14         std::string name;
15         int age;
16         std::string address;
17     };
18
19     void Student::show_data() {
20         std::cout << "num: " << num << " name: " << name << " age: " << age << " address: " << address << std::endl;
21     }
22 }
```

1_Programing Language > Cpp > Practice > HomeWork > book > 221228_3 > xt8_3_h2.h > ...

```
1 // File: xt8_3_h2.h
2 #include <iostream>
3 #include <string>
4
5 namespace student2 {
6     class Student {
7     public:
8         Student(int n, std::string nam, char s, float sco) {
9             num = n; name = nam; sex = s; score = sco;
10         }
11         void show_data();
12     private:
13         int num;
14         std::string name;
15         char sex;
16         float score;
17     };
18
19     void Student::show_data() {
20         std::cout << "num: " << num << " name: " << name << " sex: " << sex << " score: " << score << std::endl;
21     }
22 };
```

```
221228_3> g++ 221228_3.cpp -std=c++11 -o 221228_3.exe
221228_3.exe
num: 1001 name: Wang age: 18 address: 123 Beijing Road, Shanghai
num: 1102 name: Li sex: f score: 89.5
```