

# C++ 程序设计实验

---

## 实验六 静态数据和友元

2023 年 ~ 2024 年 第 1 学期

专业年级: 统计 学生学号:                      学生姓名:                     

\*\*\*\*\*

一、阅读下面程序,分析是否正确,并上机调试,指出错误的行号,并给出修改方法和原因。

```
(1)  #include<iostream>
(2)  using namespace std;
(3)  class Student{
(4)      public:
(5)      Student(string SN){
(6)          Name=SN;
(7)          Sum++;
(8)      }
(9)      static void printInfo( ) {
(10)          cout<<"The sum of students is "<<Sum<<endl;
(11)          cout<<"The student's name is "<<Name<<endl;
(12)      }
(13)  private:
(14)      string Name;
(15)      static int Sum=0;
(16)  };
(17)  void main( ) {
(18)      Student S1("Zhang San",S2("Wang Qing"));
(19)      S1.printInfo();
(20)  }
```

解答:

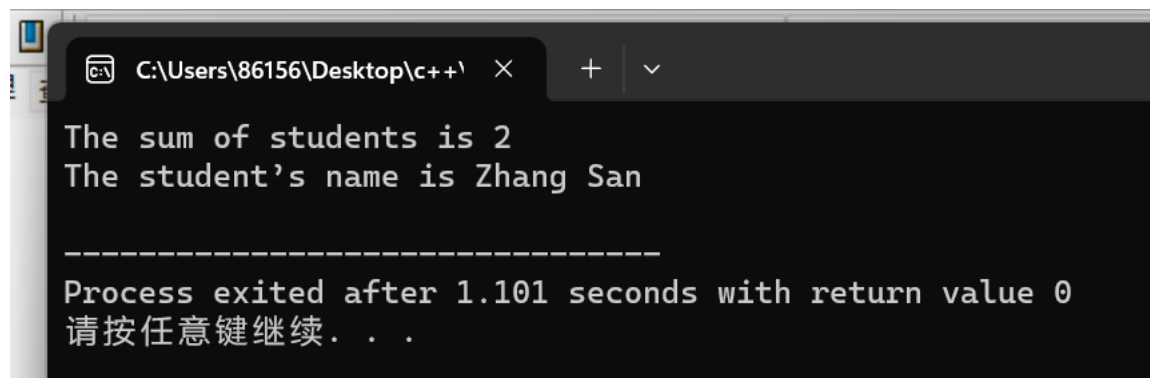
该程序存在 4 个错误。

1. (10) (11) (18) 排的引号应该使用英文的引号而不是中文的;
2. (15) 行,不能在类内初始化非 `const static` 成员,因此应该改为在类外定义 `int Student::Sum=0;`
3. (9) 行,类的静态成员函数只能访问类的静态成员变量,因此 `printInfo` 无法访问非静态成员变量 `Name`,此时选择删去 `printInfo` 函数前面的 `static`;
4. (17) 行,即使 `main` 前面的返回值定义是 `void`,最后程序编译时仍然提示 `main must return`

## C++ 程序设计实验

int; 根据查询, 并不是所有编译器都支持 void main 类型, 因此在编译器编译时改成了 int main 和 return 0; 此时显示顺利编译。

此时的运行结果为:



```
C:\Users\86156\Desktop\c++' x + v
The sum of students is 2
The student's name is Zhang San
-----
Process exited after 1.101 seconds with return value 0
请按任意键继续...
```

二、定义学生类 (Student), 数据成员包括姓名 (name)、学号 (ID)、数学成绩 (mathScore)、英语成绩 (englishScore)、总人数 (count)、数学总成绩 (mathTotalScore) 和英语总成绩 (englishTotalScore)。至少满足以下功能:

- 1) 所有数据成员为 private 型。其中, name 是 string 类型, 其它数据的数据类型为整型; 总人数、数据总成绩、英语总成绩为静态数据成员。
- 2) 构造函数通过实参传递姓名、学号、数学成绩和英语成绩来构造对象。
- 3) 构造函数亦可以通过实参传递姓名、学号创建新对象。
- 4) 定义成员函数 print 输出非静态数据成员; 定义静态成员函数 print\_total 输出静态数据成员。
- 5) 在主函数中, 定义 vector<Student> 对象, 存储多个学生的信息, 并输出学生信息以及总人数等统计信息。

### 1 题目分析

知识点: 全局变量与局部变量, 参数的传递, 类中访问权限的设置, 构造函数, 析构函数, 函数的重载, 用 vector 定义对象等。

具体分析: 本题目与实验 4 第二题的要求基本一致, 除了功能 (5) 提出了新的要求用 vector<Student> 定义对象并存储学生信息, 因此本次程序的重点也在于这里。

### 2 程序代码

```
#include<iostream>
#include<string>
```

## C++ 程序设计实验

---

```
#include<vector>
using namespace std;
class Student{
    private:
        string name;int ID,mathScore,englishScore;
        static int count,mathTotalScore,englishTotalScore;
    public:
        Student(string a,int b,int c,int d){
            name=a;ID=b;mathScore=c;englishScore=d;
            count++;mathTotalScore+=mathScore;englishTotalScore+=englishScore;}
        Student(string a,int b){
            name=a;ID=b;
            count++;mathScore=0;englishScore=0;}
        void print(){
            cout<<"学生姓名为: "<<name<<endl;
            cout<<"学号为: "<<ID<<endl;
            cout<<"数学成绩为: "<<mathScore<<endl;
            cout<<"英语成绩为: "<<englishScore<<endl;}
        static void print_total(){
            cout<<"学生总人数为: "<<count<<endl;
            cout<<"数学总成绩为: "<<mathTotalScore<<endl;
            cout<<"数学平均成绩为: "<<mathTotalScore/count<<endl;
            cout<<"英语总成绩为: "<<englishTotalScore<<endl;
            cout<<"英语平均成绩为: "<<englishTotalScore/count<<endl;}
        ~Student(){};
};
int Student::count=0;
int Student::mathTotalScore=0;
int Student::englishTotalScore=0;
int main(){
    vector<Student>Students;
    Student one("Mary",001,98,80);
    Student two("Lily",002,78,91);
    Student three("John",003);
    Students.push_back(one);
    Students.push_back(two);
```

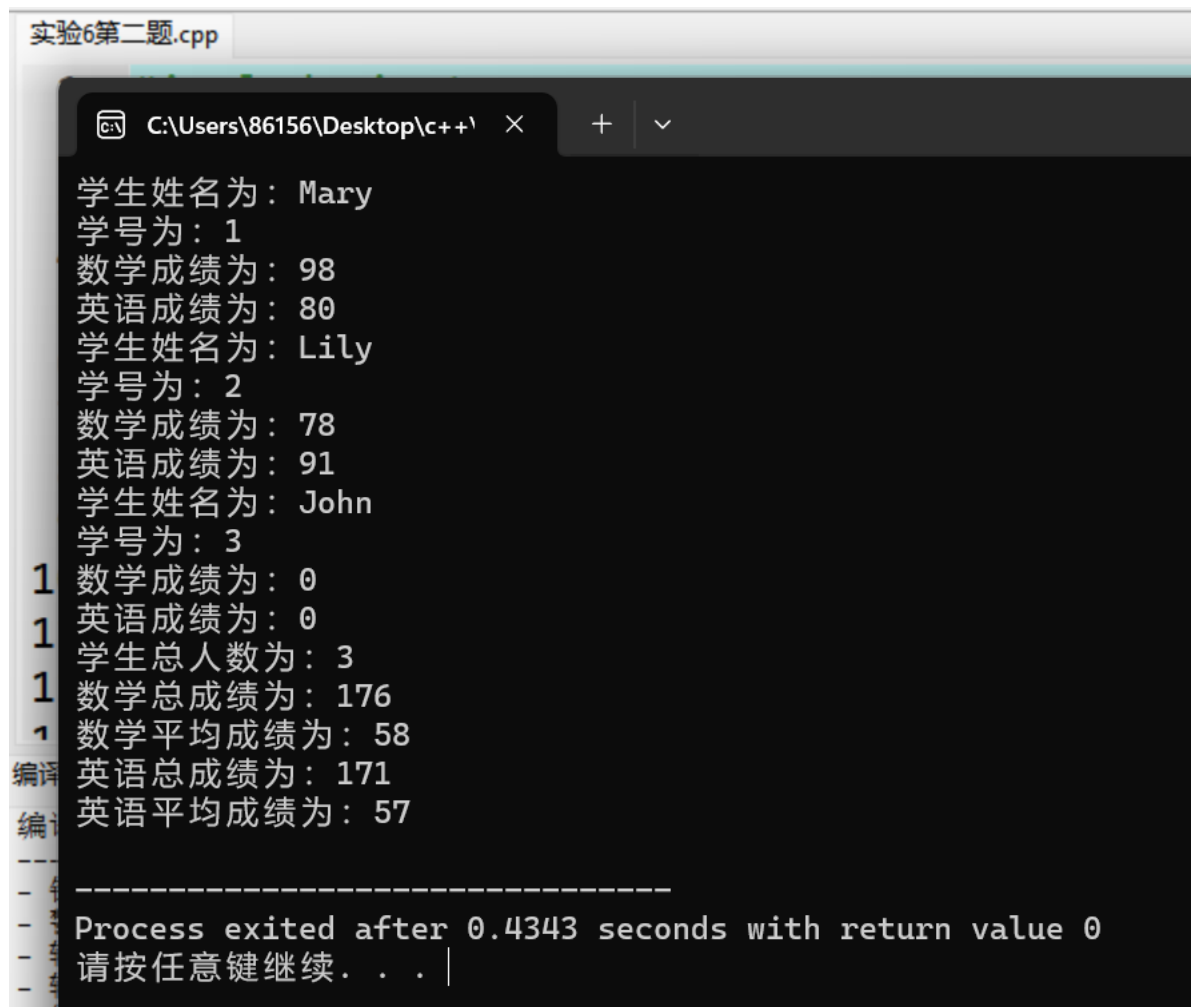
---

## C++ 程序设计实验

```
Students.push_back(three);  
for(int i=0;i<Students.size();i++){  
    Students[i].print();}  
Student::print_total();  
return 0;}
```

### 3 运行结果以及分析

结果:



The screenshot shows a Windows command prompt window titled "实验6第二题.cpp". The window displays the output of a C++ program. The output lists three students: Mary, Lily, and John, with their respective IDs and scores in Mathematics and English. It then calculates and displays the total and average scores for each subject. The program exits with a return value of 0.

```
学生姓名为: Mary  
学号为: 1  
数学成绩为: 98  
英语成绩为: 80  
学生姓名为: Lily  
学号为: 2  
数学成绩为: 78  
英语成绩为: 91  
学生姓名为: John  
学号为: 3  
1 数学成绩为: 0  
1 英语成绩为: 0  
1 学生总人数为: 3  
1 数学总成绩为: 176  
1 数学平均成绩为: 58  
编译 英语总成绩为: 171  
编译 英语平均成绩为: 57  
-----  
Process exited after 0.4343 seconds with return value 0  
请按任意键继续. . .
```

分析: 在定义完 `vector<Student>` 数组对象后, 单纯想要使用直接赋值会出现程序报错, 在课件 ppt 查找学习后, 修正了错误, 改为使用 `Student` 定义一个对象并赋值, 再使用 `push_back` 函数将定义的对象存入重新定义的 `Students` 数组中, 此时显示存入成功, 再运用循环函数对 `Students` 数组进行输出, 就可以运行程序输出结果了。