**实验五 类与对象**

**【实验目的与要求】**

1、掌握声明类的方法，类和类的成员的概念以及定义对象的方法。

2、初步掌握用类和对象编制基于对象的程序。

3、学习检查和调试基于对象的程序。

**【实验内容】**

1、有以下程序：

#include<iostream>

using namespace std；

class Time // 定义Time类

{

public： // 数据成员为公用的

int hour;

int minute;

int sec ;

};

int main()

{

Time tl; //定义t1为Time类对象

cin>>t1.hour; //输入设定的时间

cin>>t1.minute;

cin>>t1.sec;

cout<<t1.hourl<<”：”<<t1.minute<<”：”<<t1.sec<<endl ;

return 0;

}

改写程序，要求：

(1)将数据成员改为私有的；

(2)将输入和输出的功能改为由成员函数实现；

(3)在类体内定义成员函数。

然后编译和运行程序。请分析什么成员应指定为公用的?什么成员应指定为私有的? 什么函数最好放在类中定义? 什么函数最好在类外定义?

**答：**

**1. 公有成员**

**应该将那些需要被类的外部代码访问的成员指定为公有。例如，外界需要调用的方法来获取或修改对象的状态，像获取对象的属性值（如 get 方法）或者执行主要的操作函数等。**

**2. 私有成员**

**把类内部使用的数据成员和辅助函数指定为私有。这些成员是类的实现细节，不希望被外部直接访问，防止不合理的修改，例如内部数据的存储变量，仅在类内部使用的计算辅助函数等。**

**3. 类中定义的函数**

**简短的内联函数最好放在类中定义，通常是一些简单的访问器函数（ get 和 set 函数），代码简短且频繁调用，在类中定义可以提高程序的执行效率并且代码更紧凑。**

**4. 类外定义的函数**

**复杂的函数，特别是函数体较长、包含复杂逻辑或者大量代码的函数最好在类外定义。这样可以使类的声明部分更简洁，并且便于代码的维护和阅读。**

2、分别给出如下的3个文件：

(1)含类定义的头文件student.h，

//student.h (这是头文件，在此文件中进行类的声明)

class Student //类声明

{

public: //公用成员函数原型声明

void display();

private:

int num;

char name[20];

char sex;

};

(2)包含成员函数定义的源文件student.cpp

//student.cpp 在此文件中进行函数的定义

#include <iostream>

#include”student.h” //不要漏写此行，否则编译通不过

void Student::display() //在类外定义display类函数

{

cout<<”num：”<<num<<endl;

cout<<”name：”<<name<<endl;

cout<<”sex：”<<sex<<endl;

}

(3)包含主函数的源文件main.cpp。

为了组成一个完整的源程序，应当有包括主函数的源文件：

//main.cpp 主函数模块

#include <iostream> //将类声明头文件包含进来

#include “student.h”

int main()

{

Student stud; //定义对象

Student stud1(007,”tcg”,’m’);

stud.display(); //执行stud对象的display函数

return 0;

}

请完善该程序，在类中增加一个对数据成员赋初值的成员函数set\_value。上机调试并运行。

3、需要求3个长方柱的体积，请编一个基于对象的程序。数据成员包括length(长)、width(宽)、height(高)。要求用成员函数实现以下功能：

(1)由键盘分别输入3个长方柱的长、宽、高；

(2)计算长方柱的体积；

(3)输出3个长方柱的体积。

请编程序，上机调试并运行。

4、建立一个对象数组，内放5个学生的（学号，成绩），设立一个函数max，用指向对象的指针作函数参数，在max函数中找出5个学生的最高成绩者，并输出其学号。

5、设有一描述坐标点的类Point，其私有变量x和y代表一个点的(x,y)坐标值。请编写程序实现以下功能：利用构造函数传递参数，在定义对象时将x、y坐标值初始化为（60,80）；利用公有成员函数void setPoint(int i, int j)将坐标值修改为(60+i,80+j)；利用公有成员函数display()输出修改后的坐标值。主函数中通过定义对象，验证各个函数。

**一、算法分析，程序结果**

**1.**

#include<iostream>

using namespace std;

class Time // 定义Time类

{

private: // 将数据成员访问权限改为私有

int hour;

int minute;

int sec;

public:

void inp() {

cin >> hour;

cin >> minute;

cin >> sec;

} //在类体内定义用来输入的成员函数

void outp() {

cout << hour << ":" << minute << ":" << sec << endl;

} //在类体内定义用来输出的成员函数

};

int main()

{

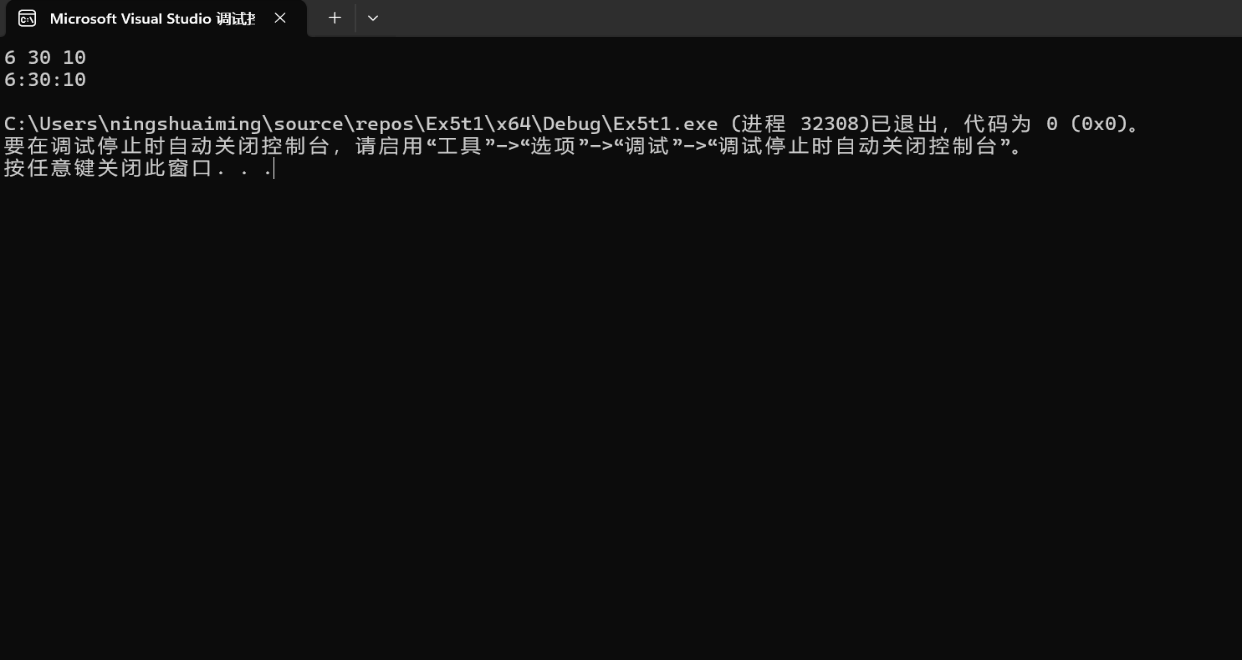
Time t1; //定义t1为Time类对象

t1.inp();

t1.outp(); //将输入和输出的功能改为由成员函数实现；

return 0;

}



**2.**

**student.h:**

//student.h (这是头文件，在此文件中进行类的声明)

#pragma once

using namespace std;

class Student //类声明

{

public: //公用成员函数原型声明

void display(); //声明一个用来输出的成员函数

void set\_value(); //声明一个对数据成员赋初值的成员函数

private:

int num;

char name[20];

char sex;

};

**student.cpp:**

//student.cpp 在此文件中进行函数的定义

#include <iostream>

#include "student.h" //不要漏写此行，否则编译通不过

using namespace std;

void Student::display() //在类外定义display类函数

{

cout <<"num："<< num << endl;

cout << "name：" << name << endl;

cout << "sex：" << sex << endl;

}

void Student::set\_value() //在类外定义set\_value()成员函数

{

cin >> num;

cin >> name;

cin >> sex;

}

**main.cpp:**

//main.cpp 主函数模块

#include <iostream> //将类声明头文件包含进来

#include"student.h"

using namespace std;

int main()

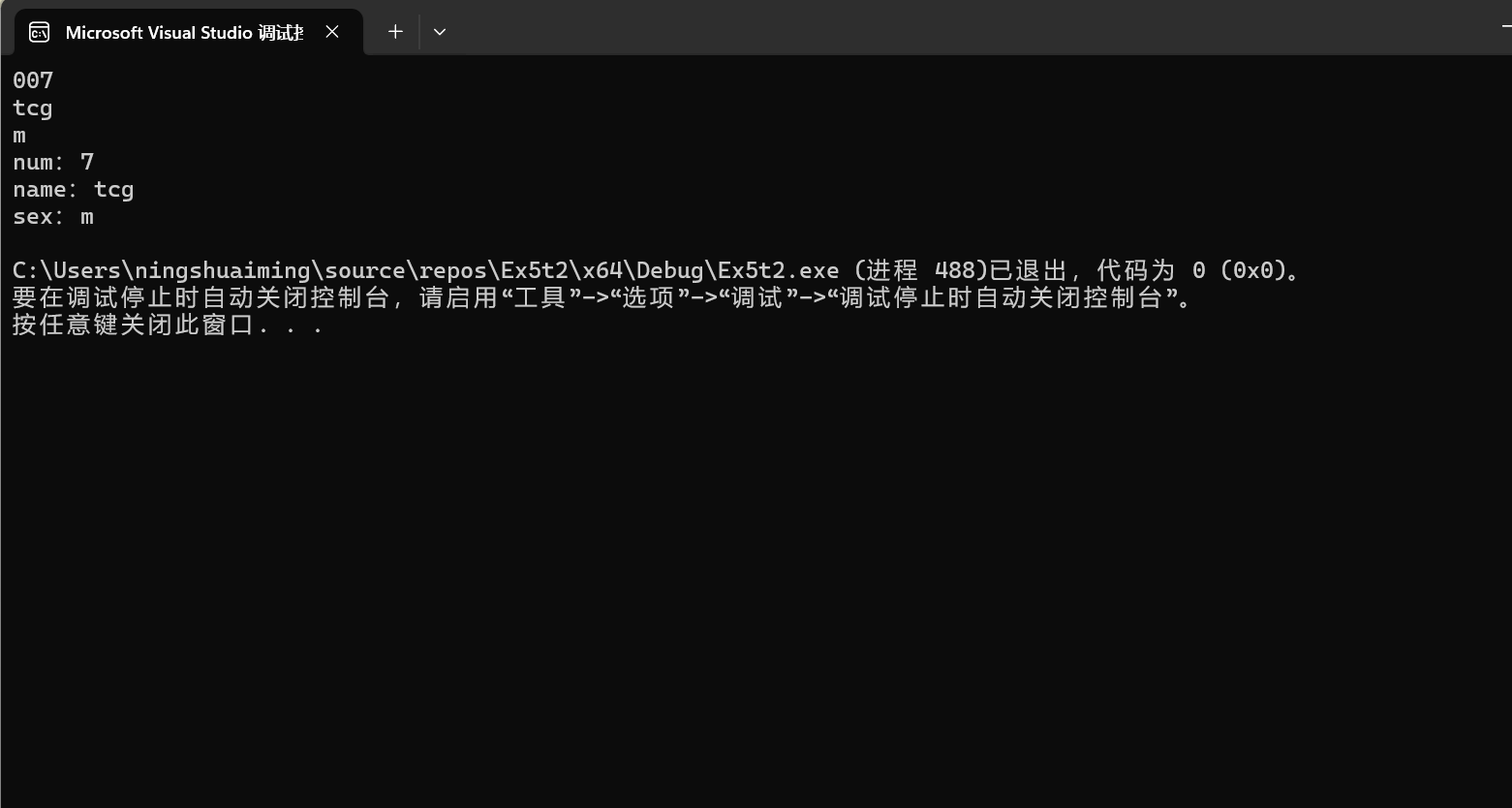
{

Student stud; //定义对象

stud.set\_value(); //执行stud对象的set\_value函数

stud.display(); //执行stud对象的display函数

return 0;

}

**3.**

**student.h:**

//student.h (这是头文件，在此文件中进行类的声明)

#pragma once

using namespace std;

class Student //类声明

{

public: //公用成员函数原型声明

void display(); //声明一个用来输出的成员函数

void set\_value(); //声明一个对数据成员赋初值的成员函数

private:

int num;

char name[20];

char sex;

};

**student.cpp:**

//student.cpp 在此文件中进行函数的定义

#include <iostream>

#include "student.h" //不要漏写此行，否则编译通不过

using namespace std;

void Student::display() //在类外定义display类函数

{

cout <<"num："<< num << endl;

cout << "name：" << name << endl;

cout << "sex：" << sex << endl;

}

void Student::set\_value() //在类外定义set\_value()成员函数

{

cin >> num;

cin >> name;

cin >> sex;

}

**main.cpp:**

//main.cpp 主函数模块

#include <iostream> //将类声明头文件包含进来

#include"student.h"

using namespace std;

int main()

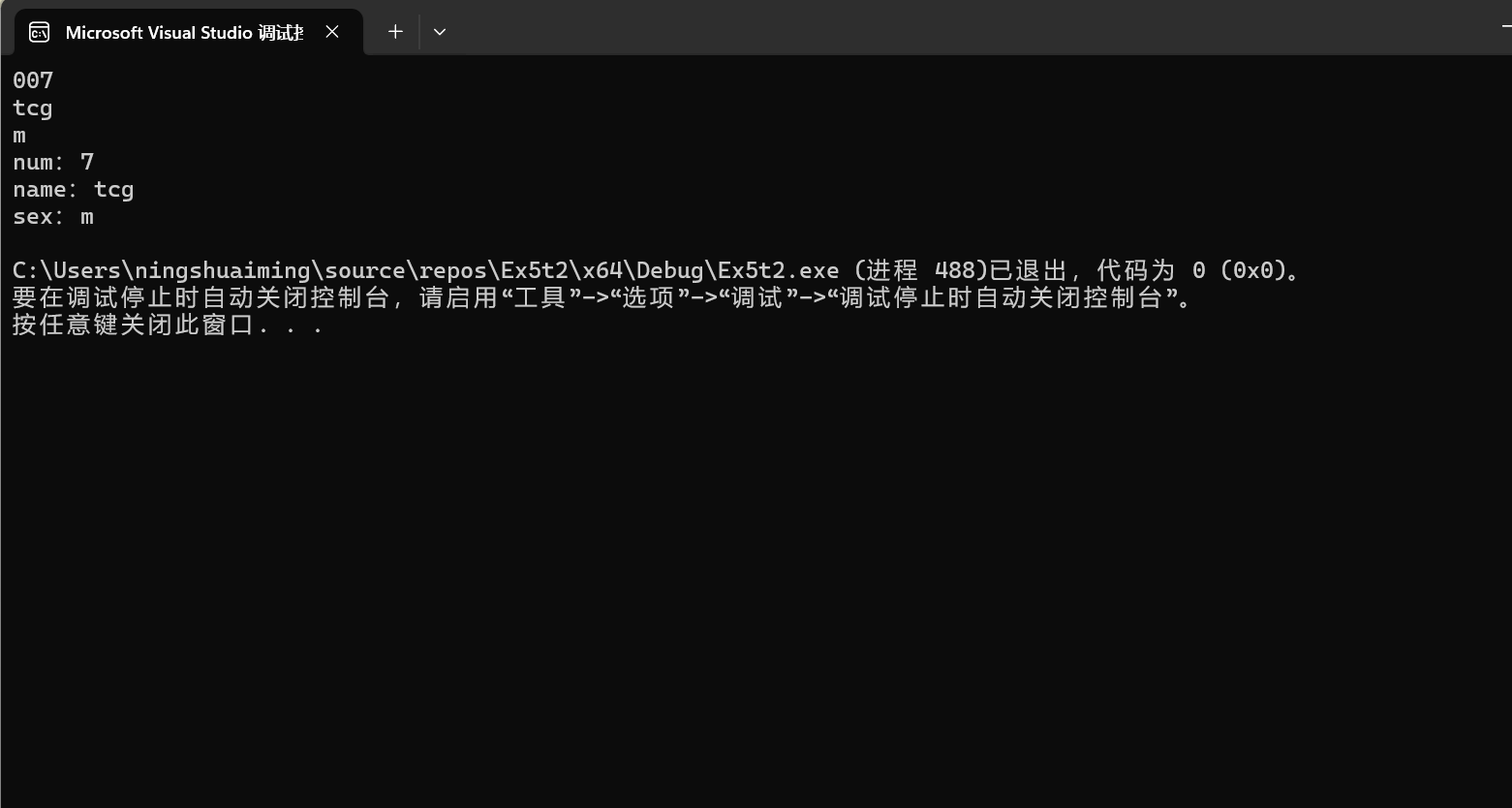
{

Student stud; //定义对象

stud.set\_value(); //执行stud对象的set\_value函数

stud.display(); //执行stud对象的display函数

return 0;

}

**4.**

#include <iostream>

using namespace std;

class Student {

friend void max(Student\* find);

private:

int num;

double grade;

public:

void inp() {

cout << "请输入该学生的学号和成绩" << endl;

cin >> num >> grade;

}

};

void max(Student\*find){ //设立一个函数max，用指向对象的指针作函数参数

bool help = false;

do {

help = true;

for (int i = 0; i < 4; i++) {

if ((find+i)->grade<(find+i+1)->grade) {

help = false;

Student mid;

mid = \*(find + i);

\*(find + i) = \*(find + i + 1);

\*(find + i + 1) = mid;

}

}

} while (help == false); //利用气泡排序的方法将成绩最高者的数据放在原对象数组的第一位，即find指针所指向的位置

cout << "成绩最高者的学号为:" << find->num << endl; //利用find指针数出成绩最高者的学号

}

int main() {

cout << "请你逐个输入五个学生的学号和成绩" << endl;

Student s1, s2, s3, s4, s5; //创建五个Student类对象

s1.inp();

s2.inp();

s3.inp();

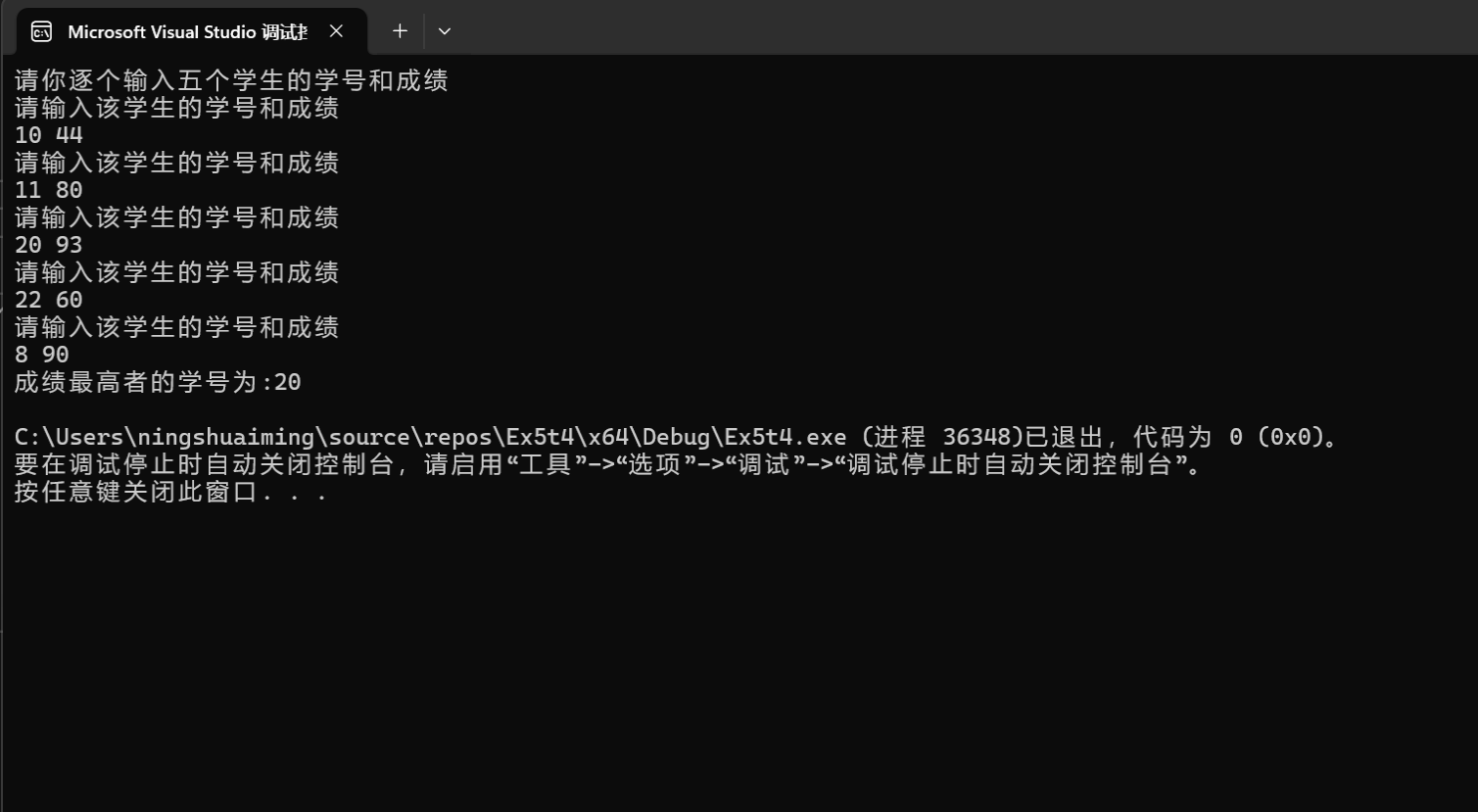
s4.inp();

s5.inp(); //为五个对象中的成员属性赋值

Student gro[5] = { s1,s2,s3,s4,s5 }; //构建一个对象数组，内放之前创建的五个对象

max(gro); //调用max函数找出5个学生的最高成绩者，并输出其学号

return 0;

}

**5.**

#include<iostream>

using namespace std;

class Point {

private:

double x;

double y; //定义私有变量x,y

public:

Point(double a,double b) {

x = a;

y = b;

} //定义有参构造函数

void setPoint(int i, int j) {

x = x + i;

y = y + j;

} //定义公有成员函数void setPoint(int i, int j)

void display() {

cout << "修改后的坐标值为：(" << x << "，" << y << ")" << endl;

} //定义公有成员函数display()输出修改后的坐标值

};

int main() {

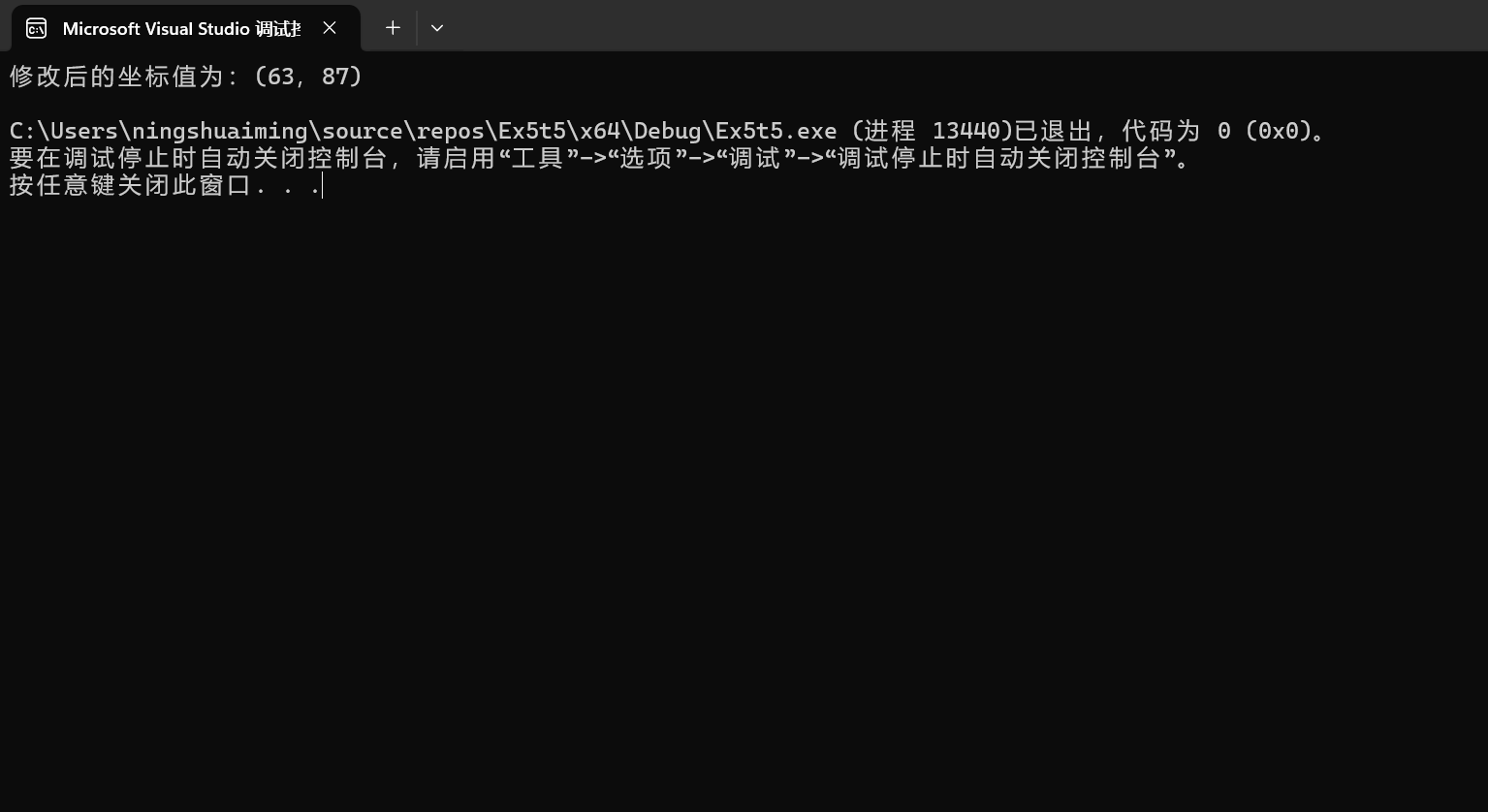
Point point1(60, 80);

point1.setPoint(3, 7);

point1.display();

return 0;

} //主函数中通过定义对象，验证各个函数



**二、遇到的问题与解决方法**

**1.** 怎么在外部函数中访问类的私有成员：

解决方法：通过复习教材，回忆起可以利用友元。即在类中声明friend void max(Student\* find)，实现对私有成员的访问。

**2.** 在仅拥有5个对象数组的首地址指针的情况下，如何找到成员属性最大的对象：

解决方法：可以利用起泡排序的思想，用如下代码：

bool help = false;

do {

help = true;

for (int i = 0; i < 4; i++) {

if ((find+i)->grade<(find+i+1)->grade) {

help = false;

Student mid;

mid = \*(find + i);

\*(find + i) = \*(find + i + 1);

\*(find + i + 1) = mid;

}

}

} while (help == false)

将该属性值最大的那个对象的所有成员放在对象数组的第一位，这样就相当于实现了 ：数组的首地址指针指向的就是该属性值最大的对象。

**三、体会**

在本次 C++实验中，我收获颇丰，深刻体会到了这门编程语言的魅力与挑战。

通过实验，我认识到温习新知识的重要性。每次实验都是对课堂所学的一次实践检验，新的语法、函数和编程逻辑若不及时复习巩固，在实际操作中就容易出现遗忘和混淆，导致错误频发，例如实验中利用友元知识访问类的私有成员。因此，对新知识的梳理和复习，能帮助我更好地理解其内涵与应用场景，为后续学习筑牢根基。

C++作为面向对象的编程语言，类与对象的熟练运用是关键。在实验过程中，我逐渐体会到类是对现实世界中事物的抽象，而对象则是这些抽象概念的具体实例。合理地设计类的成员变量与成员函数，能够清晰地组织数据和操作，使程序结构更加模块化、条理化。从创建简单的类来描述基本对象，到利用类的继承、多态等特性构建复杂的程序框架，我在不断摸索中感受到面向对象编程思想的强大表现力和灵活性，它极大地提高了代码的复用性和可维护性。

然而，实验中也暴露出我在 C++学习上的诸多不足。例如，对一些复杂的类关系理解不够深入，导致代码逻辑出现漏洞；在内存管理方面，有时会因疏忽而引发内存泄漏等问题。这让我明白，C++的学习之路任重道远，需要不断地实践与积累经验，才能真正掌握这门语言，编写出高效、健壮的程序。

本次 C++实验为我开启了一扇深入编程世界的大门，让我明确了后续学习的方向与重点，我将带着这些宝贵的经验和教训，继续在 C++的学习道路上砥砺前行。