**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?

成员函数应该指定为共有的，数据成员应该指定为私有的，类的特有的函数在类内定义，公共的函数在类外定义。

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。

**【实验流程】**

1. #include<iostream>

using namespace std;

class Time // 定义Time类

{

public:

void cinTime()

{

cin >> hour; //输入设定的时间

cin >> minute;

cin >> sec;

}

void coutTime()

{

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

}

private: // 数据成员改为私有的

int hour;

int minute;

int sec;

};

int main()

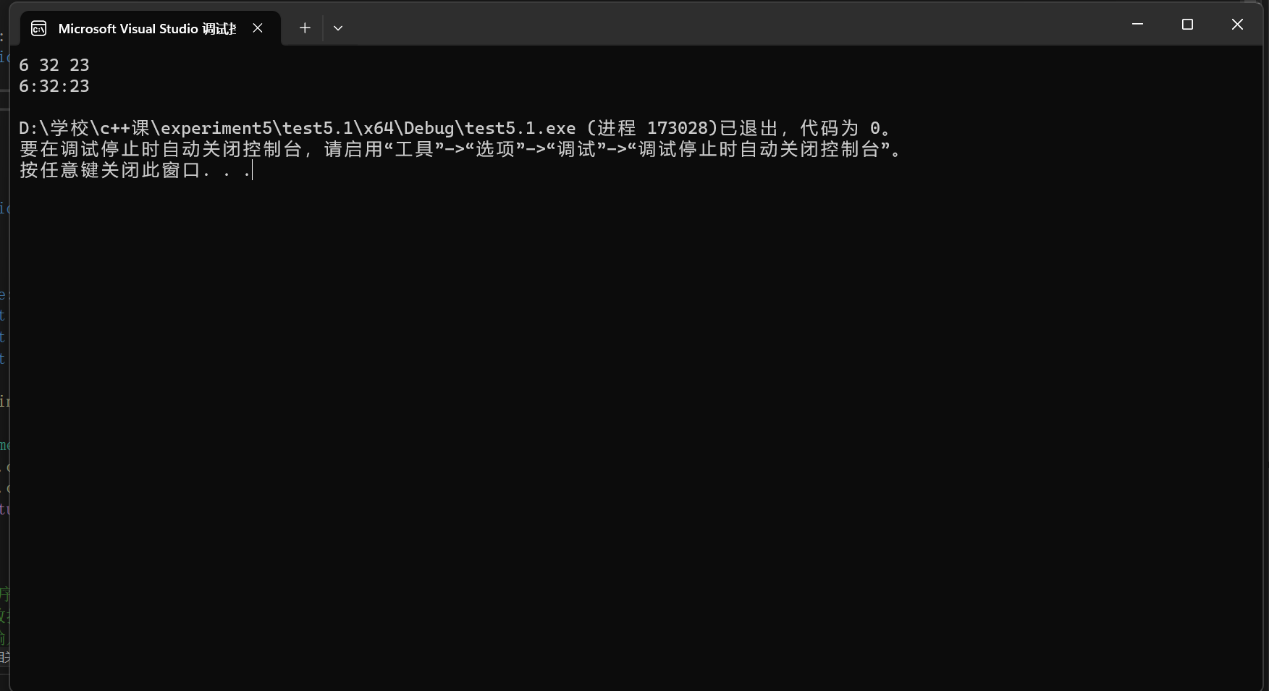
{

Time t1; //定义t1为Time类对象

t1.cinTime();

t1.coutTime();

return 0;

}

2. //student.h (这是头文件，在此文件中进行类的声明)

#include<iostream>

using namespace std;

class Student //类声明

{

public: //公用成员函数原型声明

void display();

void set\_value() //赋值成员函数

{

cout << "set name is:";

cin.getline(name, 20);//使用getline输入

cout << "set sex is:(m/f) ";

cin >> sex;

cout << "set num is:";

cin >> num;

}

private:

int num;

char name[20];

char sex;

};

//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 << "name:" << name << endl;

cout << "num:" << num << endl;

cout << "sex:" << sex << endl;

}

//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()

{

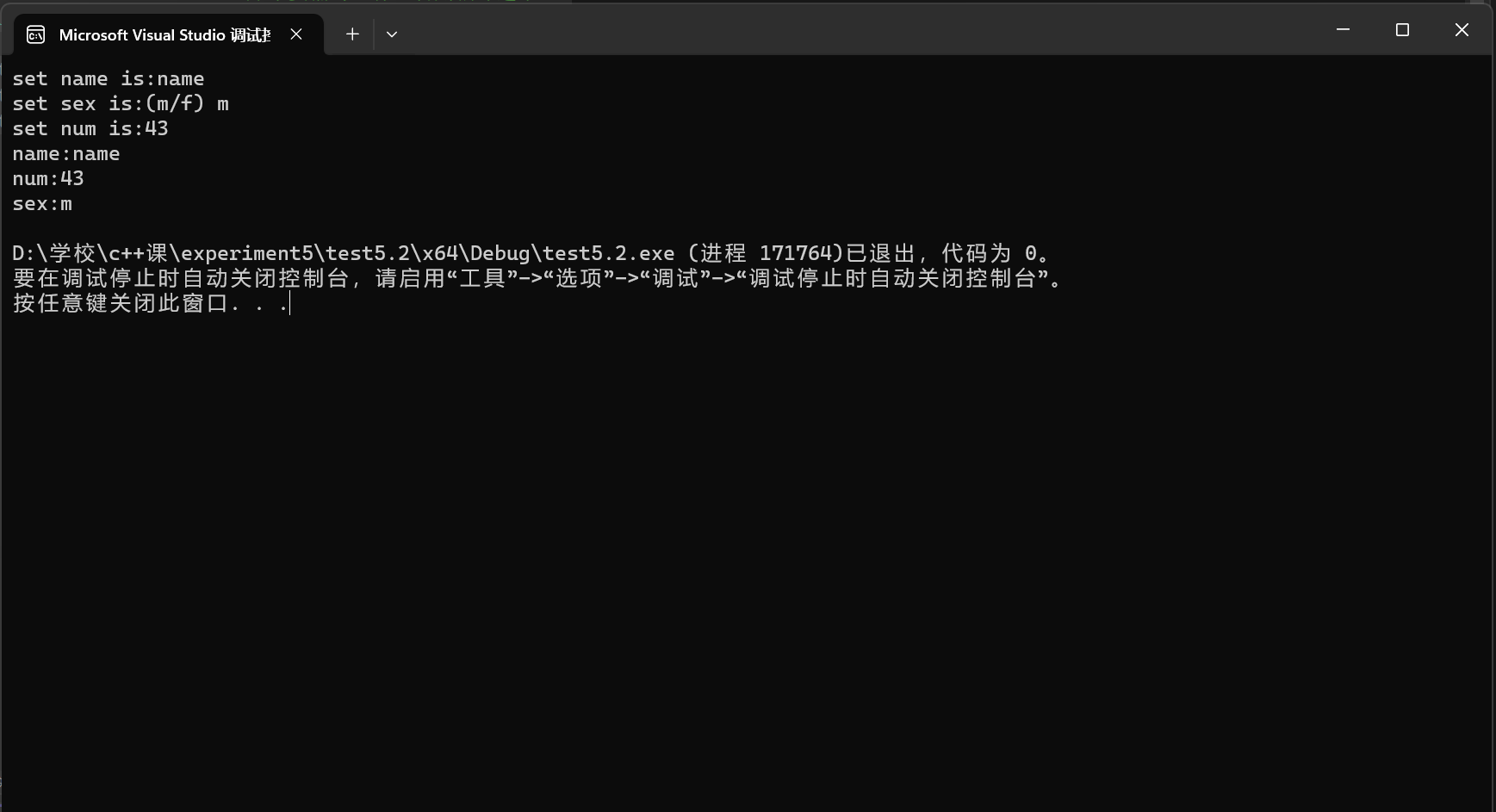
Student stud; //定义对象

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. #include<iostream>

using namespace std;

class Cuboid

{

public:

void cinCuboid()//成员函数 用键盘输入长宽高并赋值

{

cout << "请输入长方体长宽高:";

cin >> lenth;

cin >> width;

cin >> height;

}

void printV()//成员函数 计算体积并输出

{

v = lenth \* width \* height;

cout << "V=" << v << endl;

}

private:

int lenth;

int width;

int height;

int v;

};

int main()

{

Cuboid c1, c2, c3;

c1.cinCuboid();//调用赋值函数

c2.cinCuboid();

c3.cinCuboid();

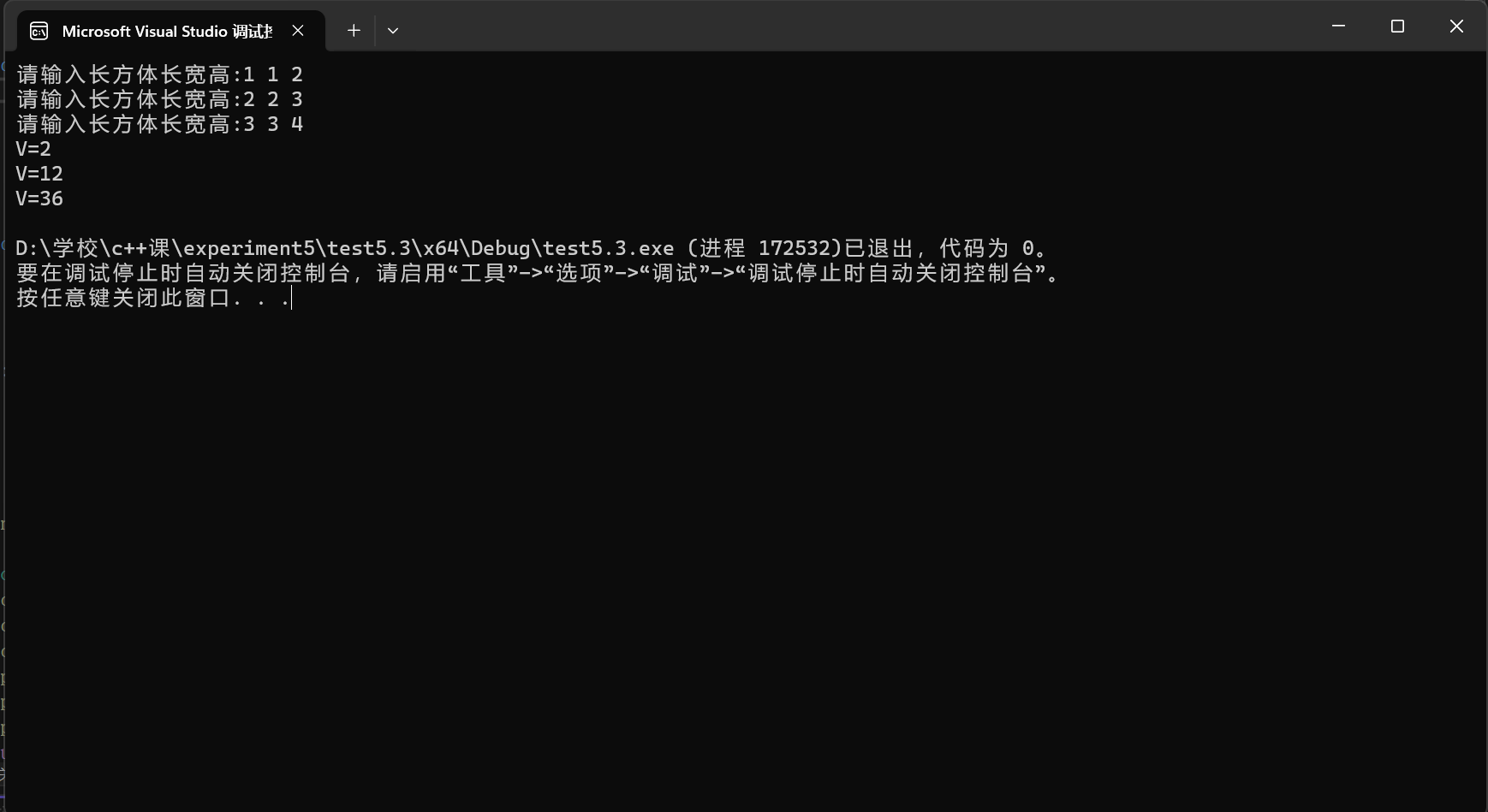
c1.printV();//调用计算输出函数

c2.printV();

c3.printV();

return 0;

}



4. #include<iostream>

using namespace std;

class Student

{

friend Student Max(Student\* stuarr);//声明友元

public:

Student(int a, int b) :id(a), mark(b){}//初始化列表

void getInformation()//输出学生信息

{

cout << id << "号学生取得" << mark << "分的最高分" << endl;

}

private:

int id;

int mark;

};

Student Max(Student\* stuarr)//定义比较最大值函数

{

Student s\_Max(0,-1);//最大值

for (int i = 0; i < 5; i++)

{

if (stuarr[i].mark > s\_Max.mark)

{

s\_Max.mark = stuarr[i].mark;

s\_Max.id = stuarr[i].id;

}

}

return s\_Max;//返回最大值

}

int main()

{

Student stuarr[5] = { //定义student类的数组

{1,20},{2,40},{3,23},{4,100},{5,96}

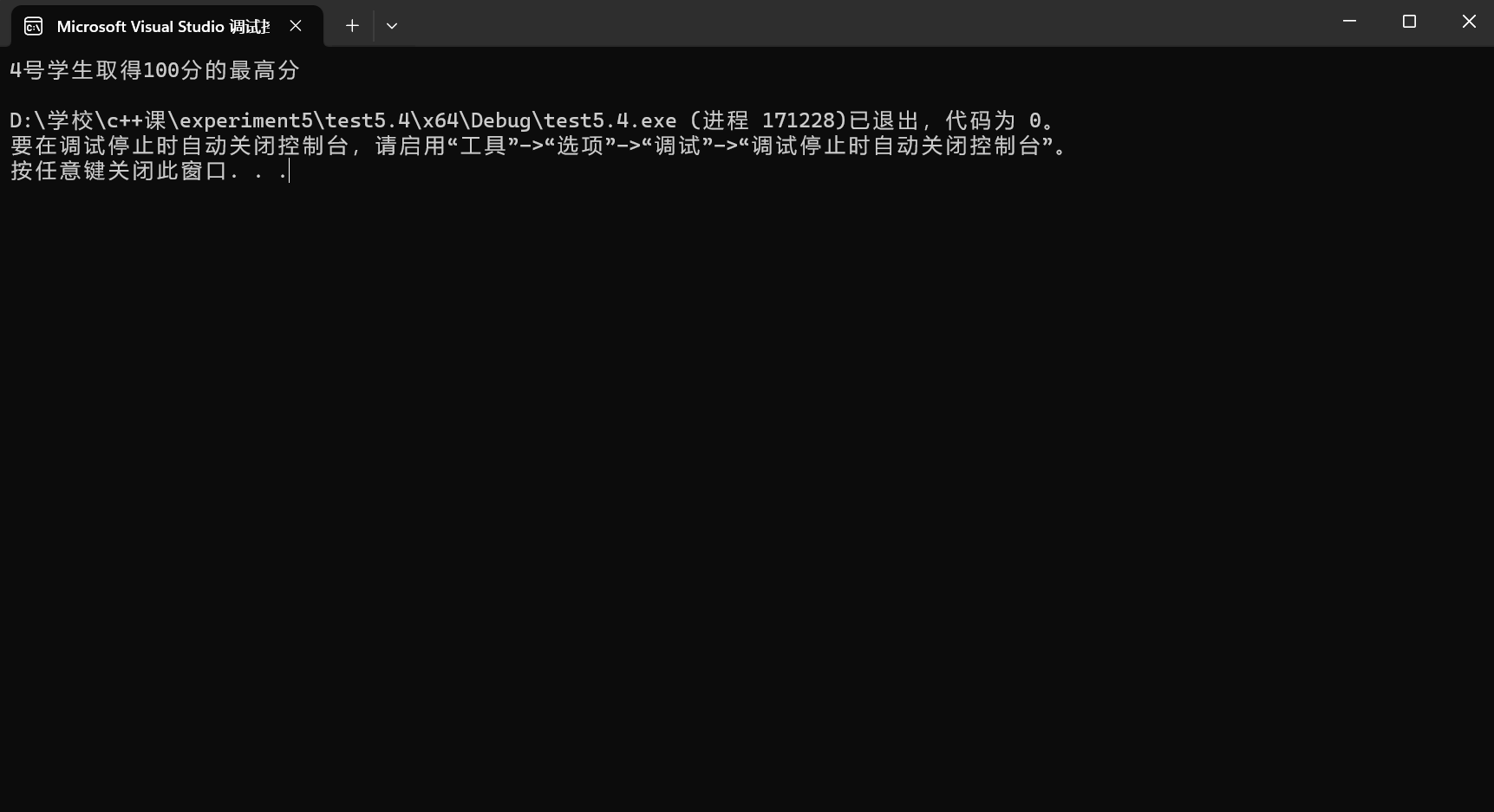
};

Student s\_max=Max(stuarr);//调用比较最大值函数

s\_max.getInformation();

return 0;

}



5. #include<iostream>

using namespace std;

class Point {

public:

Point(int a, int b)//构造函数

{

x = a;

y = b;

}

void setPoint(int i, int j)//修改点的坐标

{

x += i;

y += j;

}

void display()//输出点的坐标

{

cout << "The point is(" << x << ',' << y << ')' << endl;

}

private:

int x;

int y;

};

int main()

{

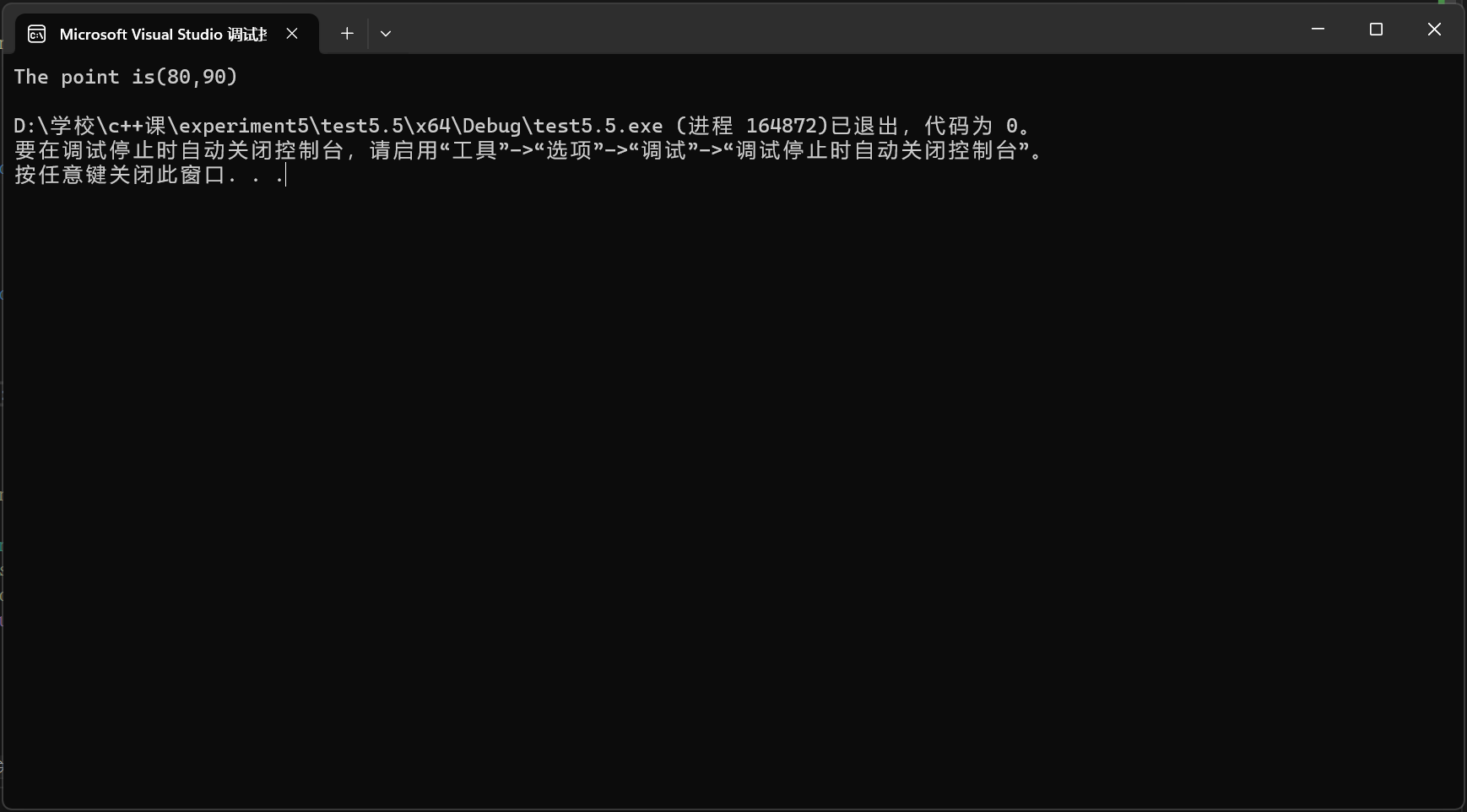
Point p1(60, 80);//定义p1点

p1.setPoint(20, 10);//修改p1点坐标

p1.display();//输出p1点坐标

return 0;

}



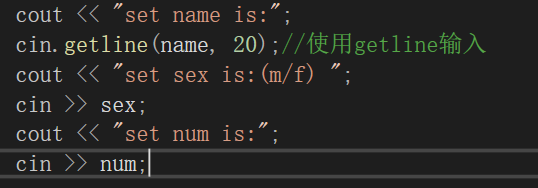
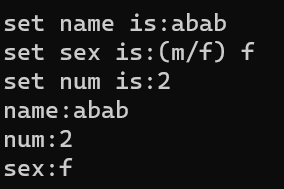
**【实验过程中遇到的问题与解决方法】**

**1.cin.getline()无法正常执行**

在进行实验5.2的过程中，需要用到cin.getline()函数，但是在使用该函数程，程序运行后，发现这一步被跳了过去。

使用f11分析，发现在进行这一步的时候无法进行输入，再次按下f11，这一步被直接跳了过去。

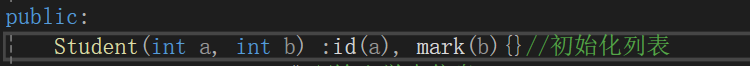
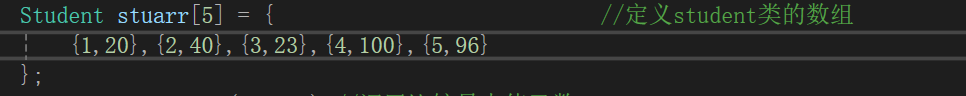
首先考虑自己是不是程序语法出现了问题，但是编译器并没有报错，且在检查之后，基本语法没有出现错误。

再考虑是不是因为将cin.getline()与cin连用，导致出现问题。在调整cin.getline()与cin的顺序后，发现程序可以正常运行，问题解决。

**2.初始化时忘了构造函数**

在实验5.4中，定义了student类的数组，在尝试对该数组进行赋值时，编译器报错。

在回想为何会报错时，才想起来自己把类当做了结构体，没有构造函数就赋值定然会报错。

 遂构造函数。

 修改后程序可以正常运行。

**【体会】**

**1.在学习时遇到理解困难的知识点时，可以通过实践的方式理解知识。**

在尝试自学类与对象的内容时，我对于课本上、文档中给出的定义以及解释不能很好地理解，在反复研读了多次后，对于这一块的知识仍不存在多少印象，也不会使用，对于知识十分地不熟悉。

为了加深对知识的理解，我尝试将书上、文档中的示例代码自己实现，在自己敲了一遍后，发现对知识确实更加理解了，也对如何运用这一知识有了更加深刻的认知。与此同时也认识到了学习理论知识要与实践相结合的重要性。

**2.类与对象的知识非常重要，要可以灵活掌握使用。**

C++语言是面向对象的编程语言，C++面向对象的三大特性是封装、继承、多态。

类与对象的知识在C++中有着十分重要的作用，且这里的知识相对于前面所学的知识更加复杂、丰富，如果可以掌握类与对象的相关知识，并能够灵活运用，就能够在编程过程中将C++的特点发挥出来了。