**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?

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。

1. 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】**

**五．1**

#include<iostream>

using namespace std;

class Time // 定义Time类

{

private: // 数据成员为公用的

int c\_hour;

int c\_minute;

int c\_sec;

public:

int gethour(int hour)//定义成员函数并把私有的值return 出来使之可以访问

{

c\_hour = hour;

return c\_hour;

}

int getminute(int minute)

{

c\_minute = minute;

return c\_minute;

}

int getsec(int sec)

{

c\_sec = sec;

return c\_sec;

}

};

int main()

{

Time t1; //定义t1为Time类对象

int hour, hour1;

int minute, minute1;

int sec, sec1;

cin >> hour; //输入设定的时间

cin >> minute;

cin >> sec;

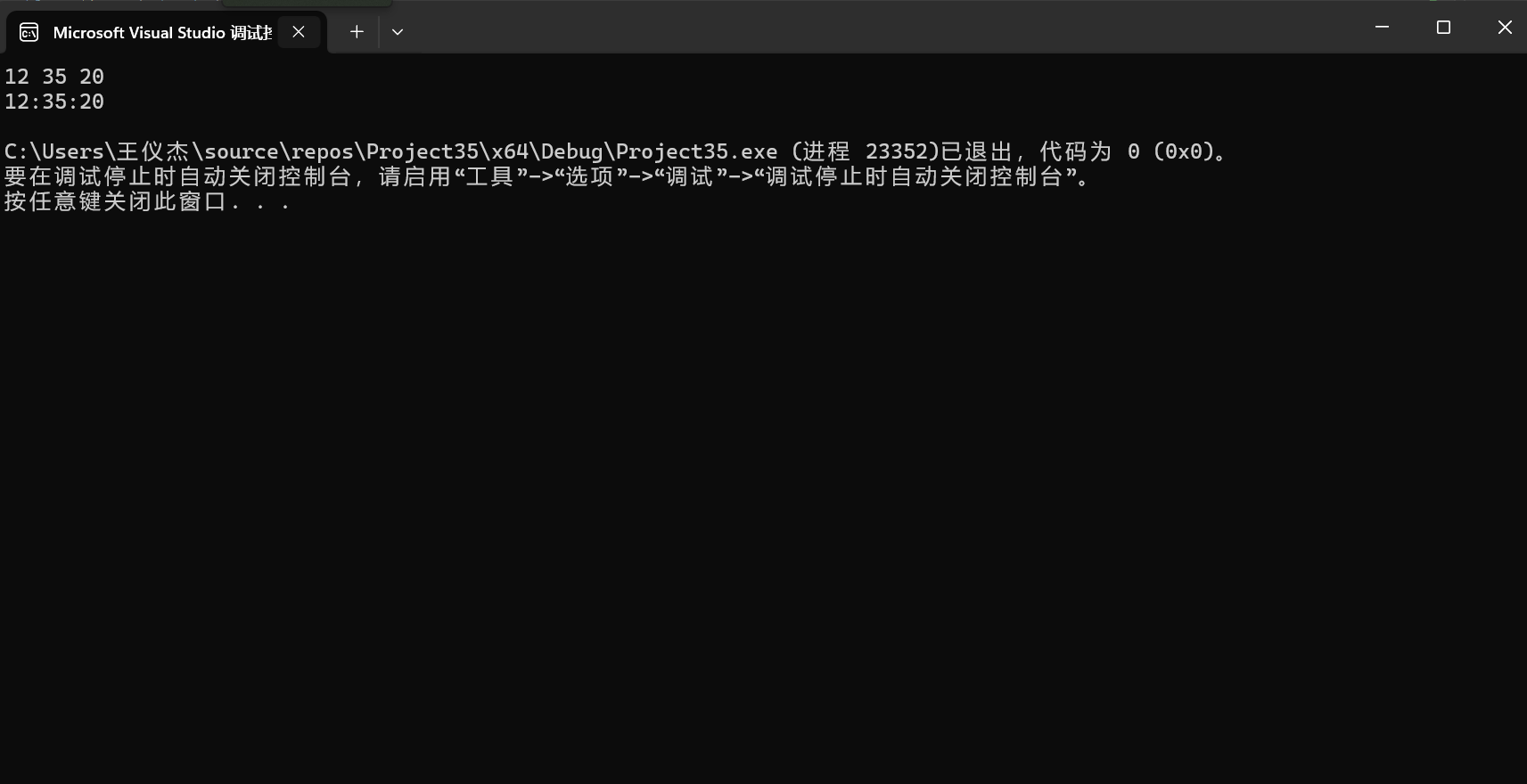
hour1 = t1.gethour(hour);//调用成员函数并且输出

minute1 = t1.gethour(minute);

sec1 = t1.getsec(sec);

cout << hour1 << ":" << minute1 << ":" << sec1 << endl;

return 0;



**五．2**

**Student.h头文件**

#include<iostream>

using namespace std;//可以统一将类定义在头文件中

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;

public:

void set\_value(int s\_num,const char s\_name[20],char s\_sex)

{

num = s\_num;

for (int i = 0; i < 20; i++)//利用循环对数组进行赋值，这里用cin.getline(name,20)进行赋值也可以

{

name[i] = s\_name[i];

}

sex = s\_sex;

}

};

**Student.cpp源文件**

//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,注意引用头文件要加引号 .h头文件中写过函数的基本框架后在cpp文件中就不用写了

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

{

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

cout <<"name:" << name << endl;

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

}

**Main.cpp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()

{

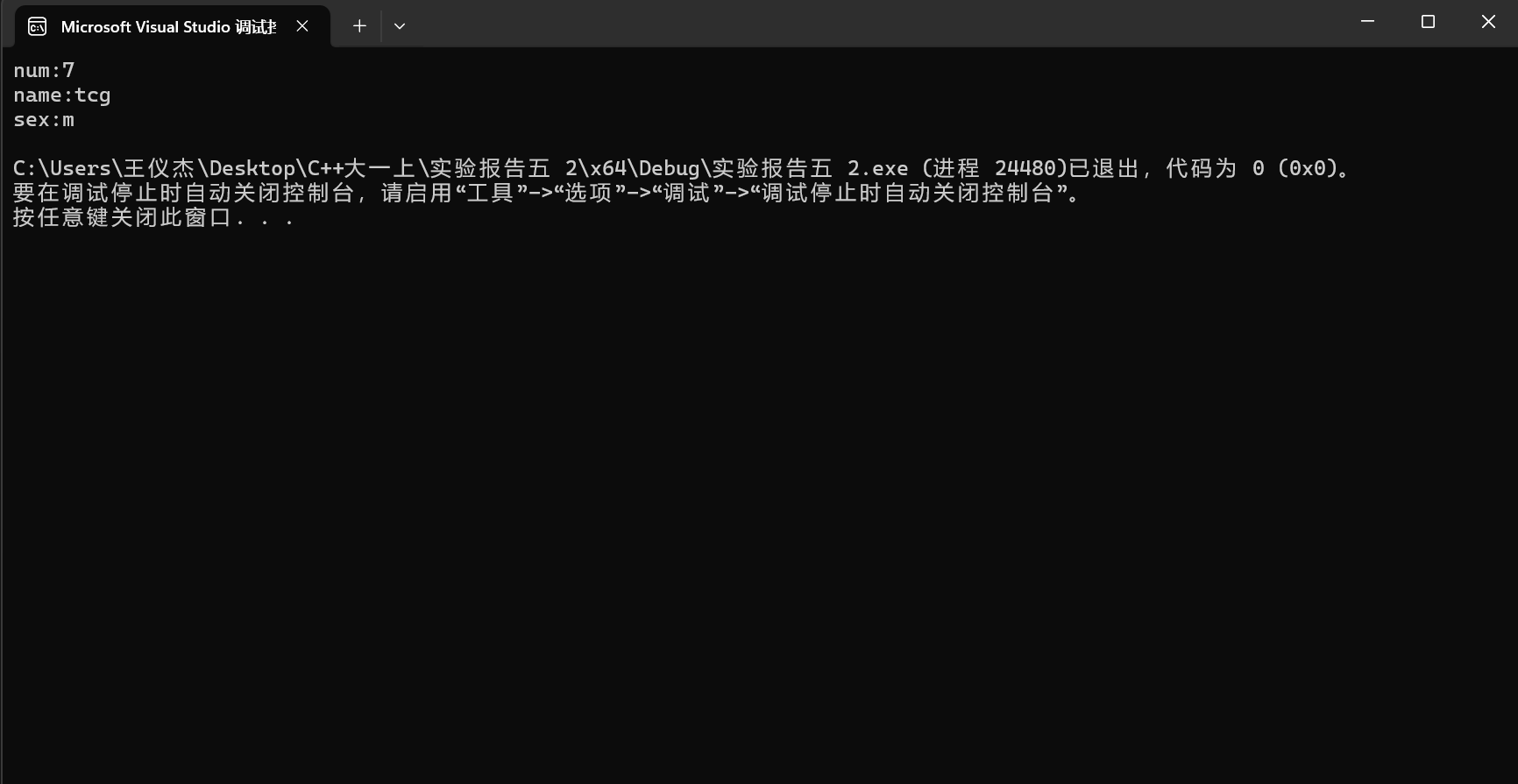
Student stud; //定义对象

stud.set\_value(007, "tcg",'m');//主函数中进行函数的调用

stud.display(); //执行stud对象的display函数

return 0;

}



**五．3**

#include<iostream>

using namespace std;

class cuboid//长方体类

{

public:

int length;

int width;

int height;

int volume()

{

int sum = length \* width \* height;

return sum;

}

};

int main()

{

cuboid cuboid1;//这三个逐个进行输入 并计算体积即可

cin >> cuboid1.length >> cuboid1.width >> cuboid1.height;

int sum1 = cuboid1.volume();

cuboid cuboid2;

cin >> cuboid2.length >> cuboid2.width >> cuboid2.height;

int sum2 = cuboid2.volume();

cuboid cuboid3;

cin >> cuboid3.length >> cuboid3.width >> cuboid3.height;

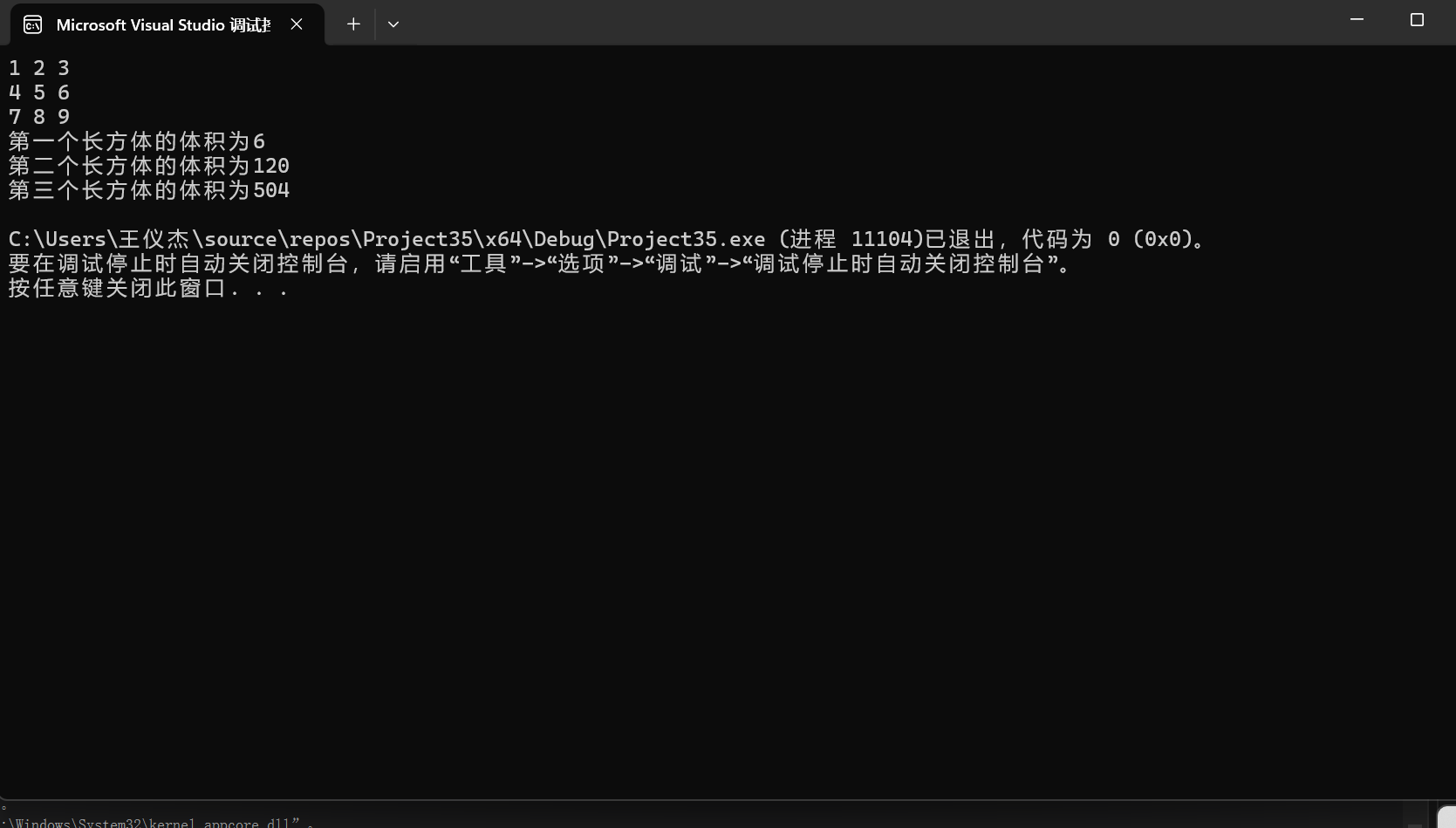
int sum3 = cuboid3.volume();

cout << "第一个长方体的体积为" << sum1 << endl;

cout << "第二个长方体的体积为" << sum2 << endl;

cout << "第三个长方体的体积为" << sum3 << endl;

}



**五．4**

#include<iostream>

using namespace std;

class Student

{

public:

int id;

int mark;

public:

Student(int a, int b)//利用构造函数进行初始化

{

id = a;

mark = b;

}

void getInformation()//输出学生信息

{

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

}

};

Student Max(Student\* arr)//定义比较最大值函数

{

Student max\_student(0, 0);//括号法调用构造函数id 和mark都初始化为0

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

{

if (arr[i].mark > max\_student.mark)//利用循环取到最大值

{

max\_student.mark = arr[i].mark;

max\_student.id = arr[i].id;

}

}

return max\_student;//返回最大值

}

int main()

{

Student arr[5] =

{//定义student类的数组

{01,100},

{02,99},

{03,98},

{04,97},

{05,96}

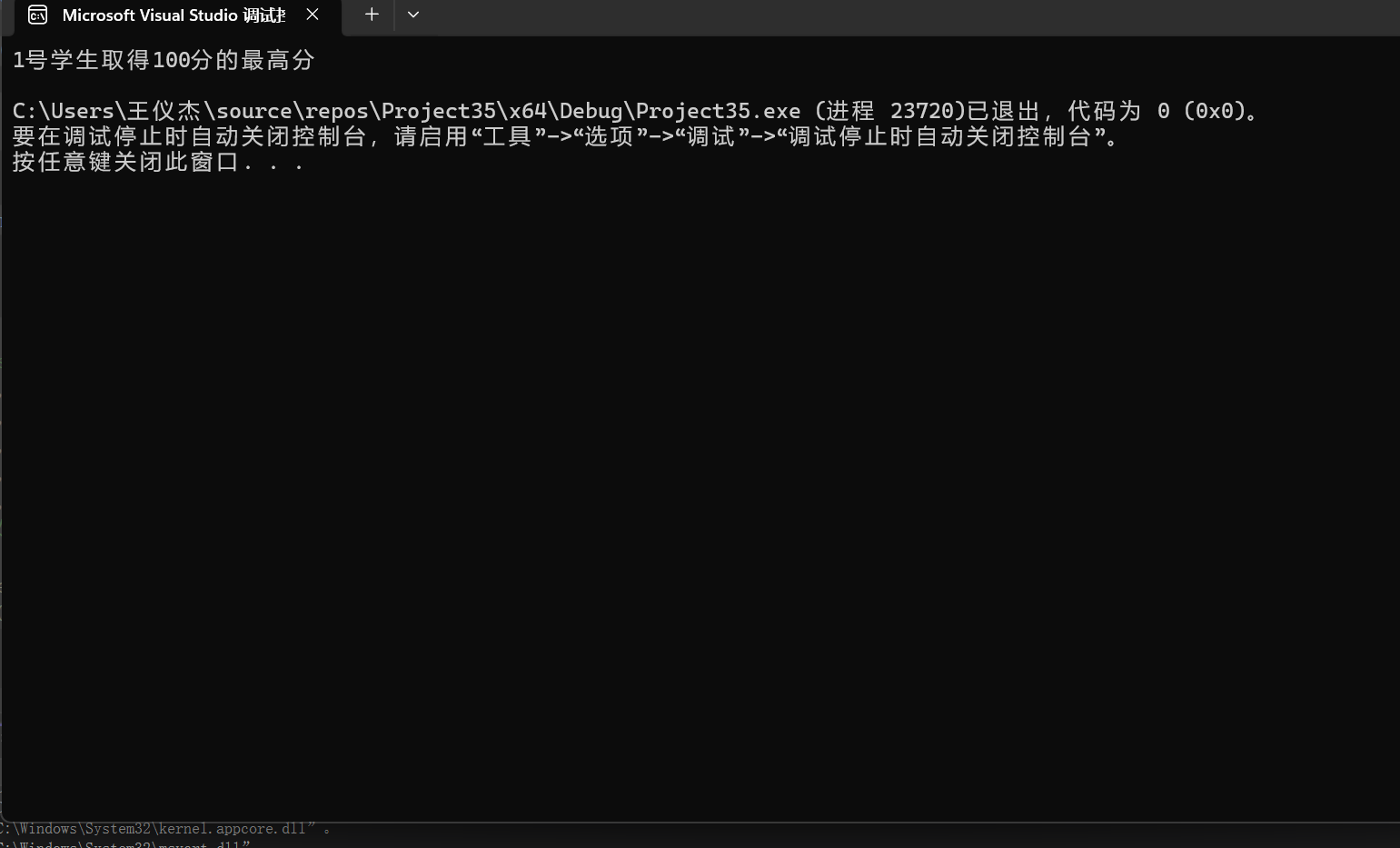
};//定义这样的student类型的数组时，可以直接采用arr[i].id的方式来访问类中的东西 这个就叫对象数组

Student s\_max = Max(arr);//调用比较最大值函数

s\_max.getInformation();

return 0;

}



**五．5**

#include<iostream>

using namespace std;

class point

{

private:

int x;

int y;

public:

point()//默认构造函数赋初值

{

x = 60;

y = 80;

}

void setpoint(int i, int j)//成员属性为私有的不能调用，但是如果作为函数的返回值或在函数中直接cout那么可以调用，这就是成员属性私有的作用

//不然设置为私有又不能访问，没有一点作用，只能通过函数来对其进行赋值

{

x = 60 + i;

y = 80 + j;

}

void display()

{

cout << "(" << x << "," << y << ")" << endl;

}

};

int main()

{

point p;

int i;

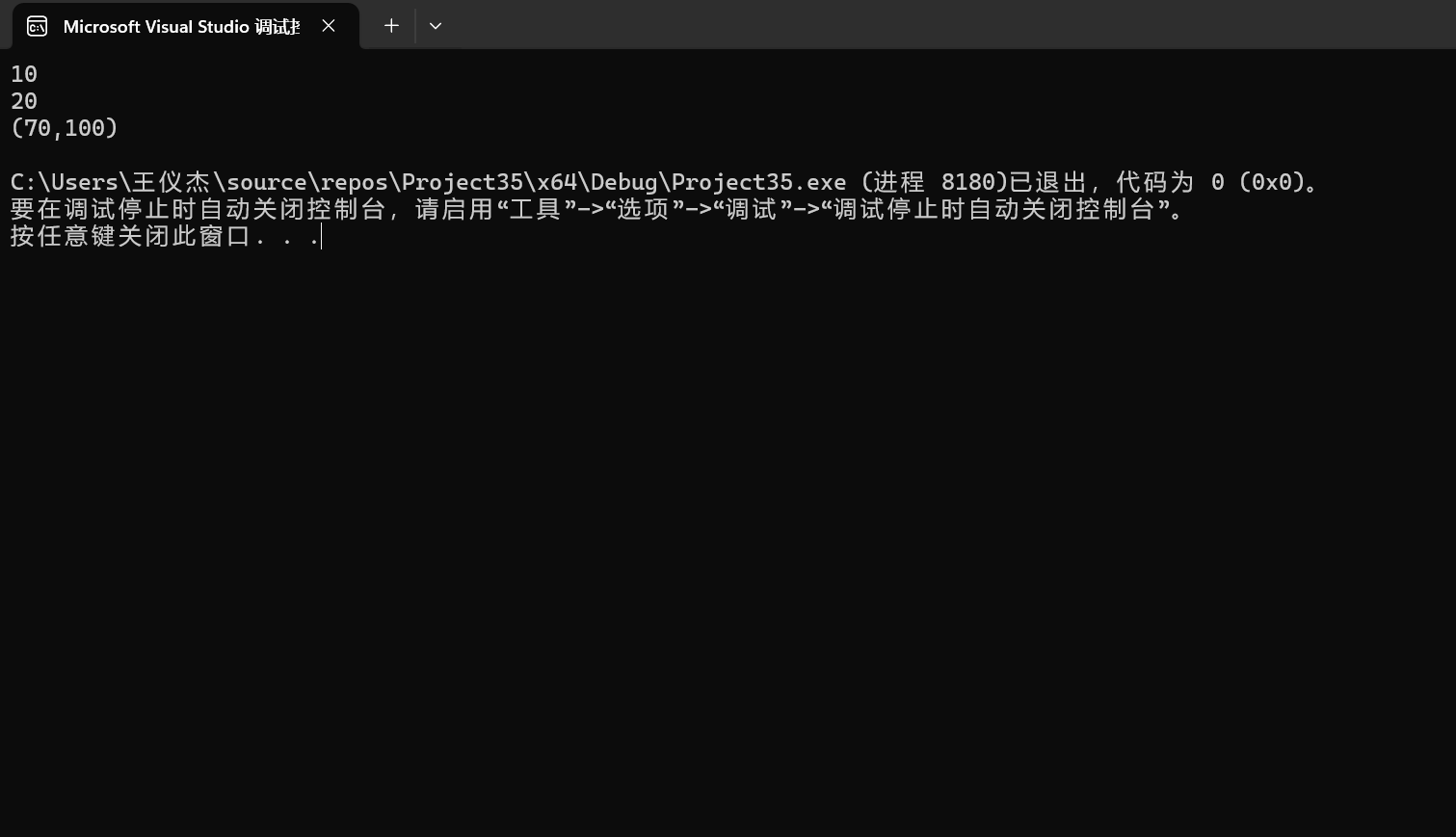
int j;

cin >> i >> j;

p.setpoint(i, j);//默认构造函数会自动调用，这里调用公共成员函数

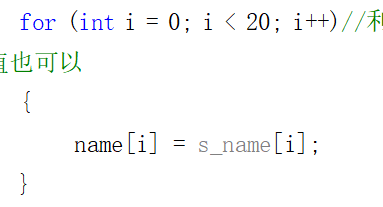
p.display();

**}**

****

**【实验过程中遇到的问题与解决方法】**

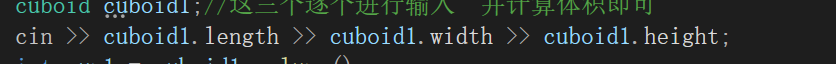
**五．2**

****

**这一步通过for循环对name进行赋值，也可以通过cin.getline(name,20)来进行赋值操作。**

**但是要注意cin.getline不能于cin>>连用**

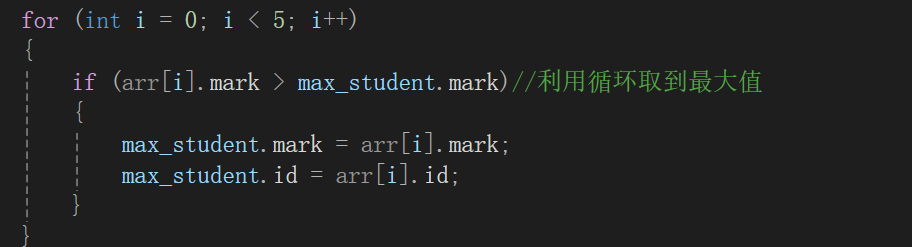
**五．3**

****

**要注意只要是对类中的东西进行访问，那么一定要加上.进行访问不要忘记了**

**五.4**

**在设置类的时候，我忘记定义构造函数对其进行赋初值操作了。定义出构造函数并将类中的成员变量进行初始化，这样在后面定义取最大值的函数时才能更容易利用for循环进行比较并赋值。**

****

**五．5**

**默认构造函数的作用就是为成员变量赋初值**

**也可以有另外一种写法**

**Student():成员变量1（值1），成员变量2（值2）**

**【体会】**

1. **要增加对类与对象操作的熟练度**

**类与对象的内容较为固定且语法较为简单，但是有更多的细节问题需要注意，因此增加练习次数以及熟练度，在进行多次实验之后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。**

1. **要更加注重各种细节问题**

**在类与对象这一内容中，有很多需要注意的细节问题，比如构造函数用于给对象赋初值并且会由系统自动调用而非进行手动调用，有参构造函数和拷贝构造函数需要自己手动调用，并且要注意那三种调用方式。其次要理解构造函数的调用规则，理解什么时候系统会提供默认构造函数和拷贝构造函数以及构造函数的调用规则。**

1. **需要强化对知识的理解**

**在自学这一方面的内容时，对一些概念的理解不到位，经过我将书上的代码进行一一实现，加强了我对类与对象这一方面内容的理解，对于知识的运用也更加的熟悉。**