**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。

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。

**算法分析，程序结果**

1.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | #include<iostream>  using namespace std;  class Time                // 定义Time类  {  public:      void input(int a,int b,int c) //输入函数      {          this->hour = a;          this->minute = b;          this->sec = c;      }        void show()    //输出函数      {          cout << "输入的时间为：" << endl;          cout <<hour << ": " << minute << ": " << sec << endl;      }  private:      int hour;   //定义时间      int minute; //定义分钟      int sec;    //定义秒钟  };  int main()  {      Time t1;      t1.input(12, 30, 26);      t1.show();      return 0;  } | |



我认为常常被类外调用的成员和函数应该定义为公有，而隐私类的不想他人访问的设置为私有。

函数比较小的或者构造函数和析构函数放在类内，而函数较大的应放在类外

2.

头文件

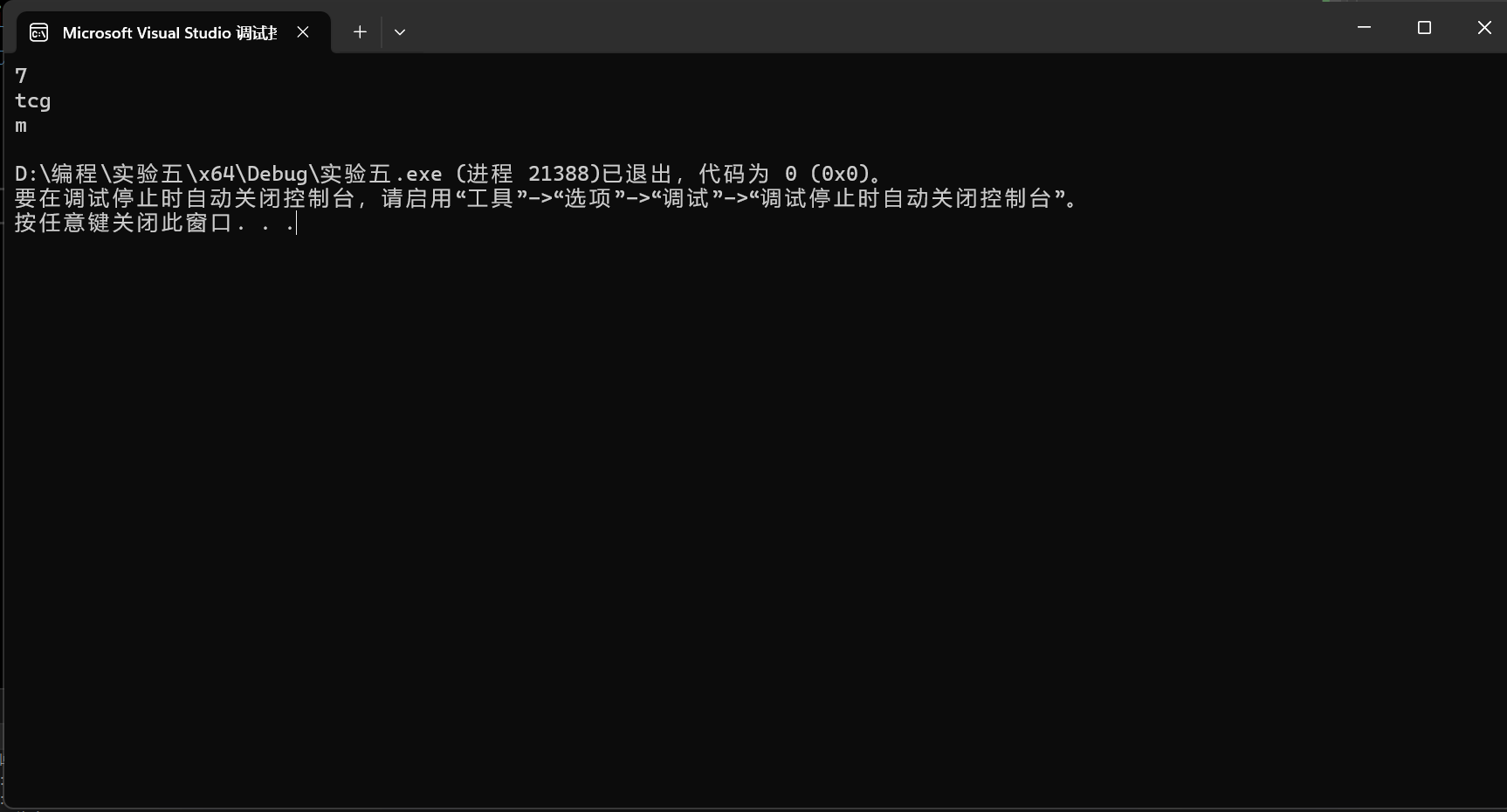
|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | #pragma once  #include <string>  class Student              //类声明  {  public:           //公用成员函数原型声明        void set\_value(int n, const char\* nm, char s);      void display();    private:      int num;      char name[20];      char sex;  };        void show()    //输出函数      {          cout << "输入的时间为：" << endl;          cout <<hour << ": " << minute << ": " << sec << endl;      }  private:      int hour;   //定义时间      int minute; //定义分钟      int sec;    //定义秒钟  };  int main()  {      Time t1;      t1.input(12, 30, 26);      t1.show();      return 0;  } | |

student.cpp

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | #include <iostream>  #include"student（实验五.2）.h"  //不要漏写此行，否则编译通不过  using namespace std;  void Student::set\_value(int n, const char\* nm, char s)  {      num = n;      for (int i = 0; i < 19 && nm[i] != '\0'; i++)      {          name[i] = nm[i];      }      name[19] = '\0';      sex = s;  }void Student::display()         //在类外定义display类函数  {      cout << "num：" << num << endl;      cout << "name：" << name << endl;      cout << "sex：" << sex << endl;  } | |

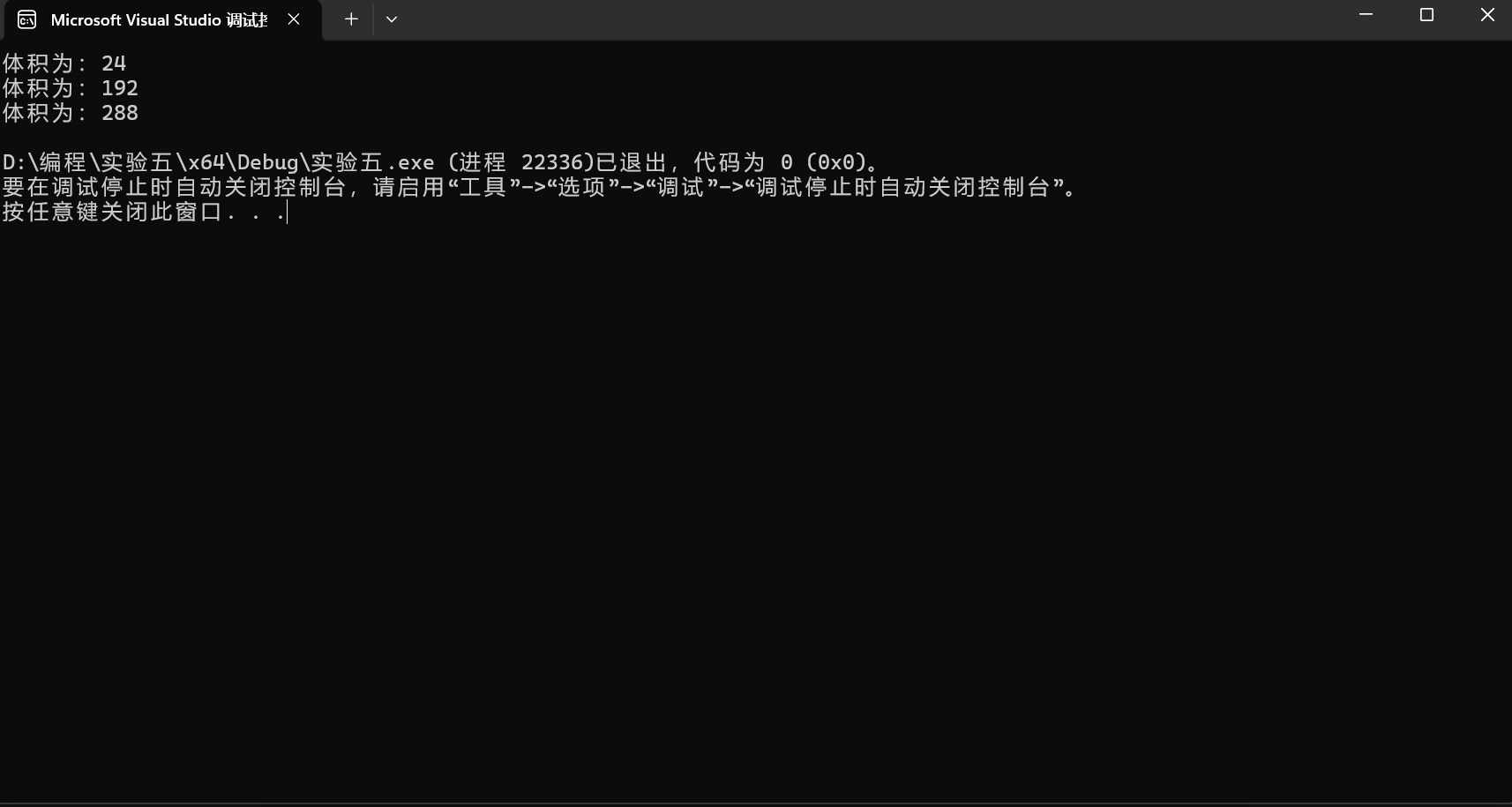
main.cpp

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11 | #include<iostream>  using namespace std;  #include"student（实验五.2）.h"  int main()  {      Student stud;                     // 定义对象      stud.set\_value(007, "tcg", 'm');  // 调用 set\_value 设置值      stud.display();                   // 调用 display 显示数据        return 0;  } | |



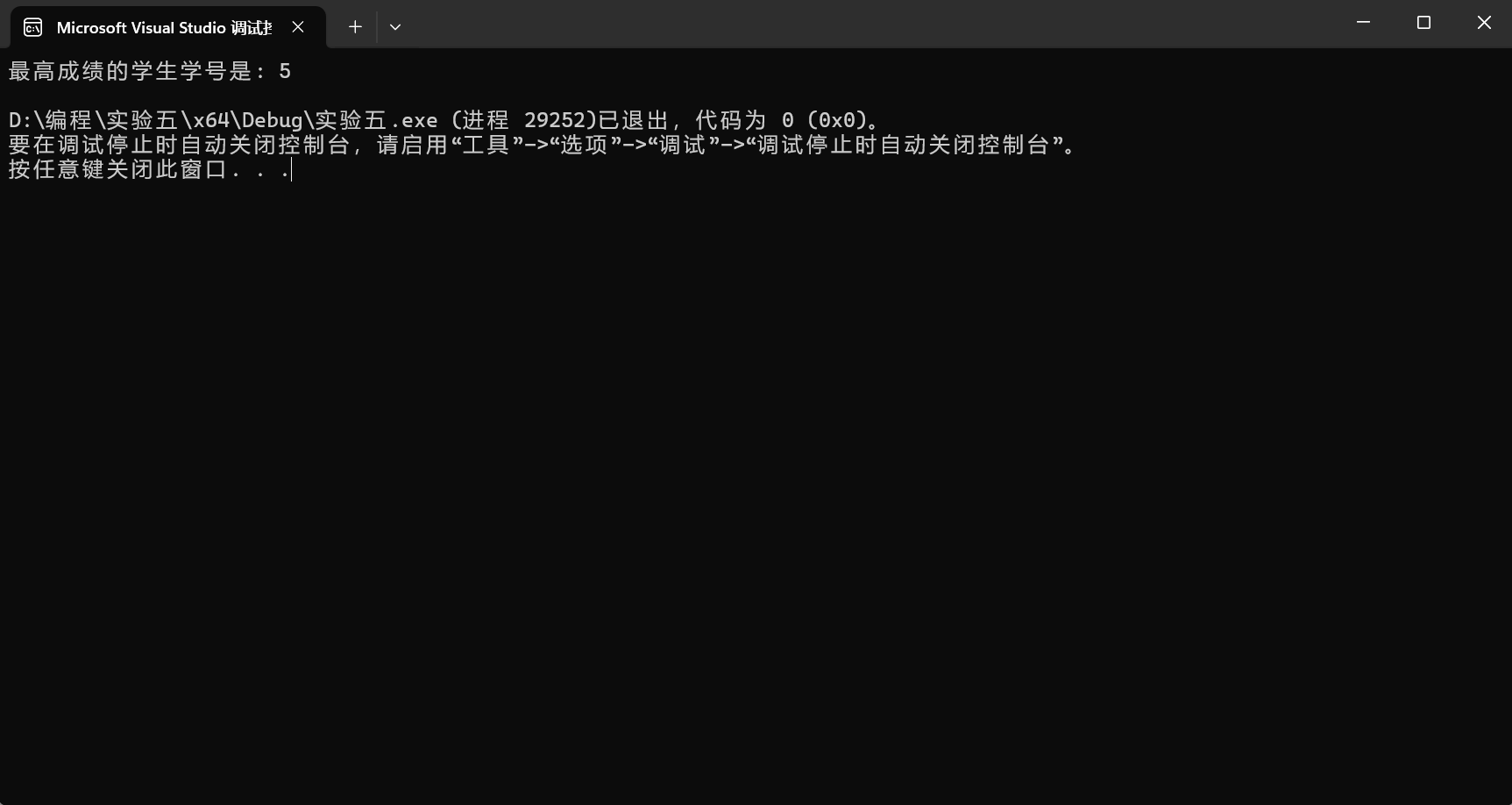
3.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | #include<iostream>  using namespace std;  class cft  {  public:      void input(int a, int b, int c)//输入函数      {          this->length = a;          this->width = b;          this->height = c;          v = a \* b \* c;      }      void show()//输出函数      {          cout << "体积为：" << v << endl;      }  private:      int length;//长      int width;//宽      int height;//高      int v;//体积  };  int main()  {      cft t1;      cft t2;      cft t3;      t1.input(2, 3, 4);      t2.input(4, 6, 8);      t3.input(6, 8, 6);      t1.show();      t2.show();      t3.show();      return 0;  } | |



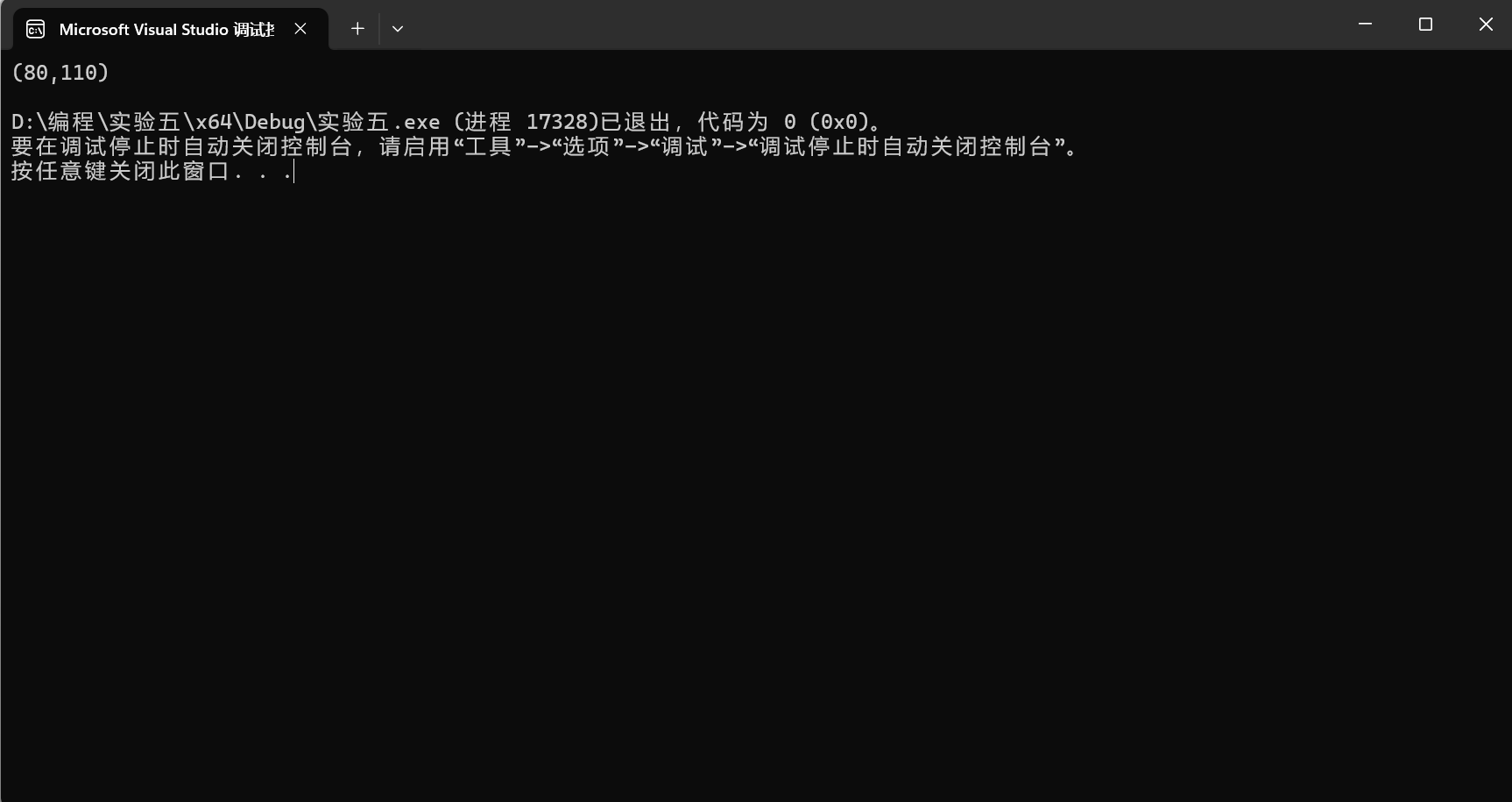
4.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | #include <iostream>  using namespace std;    // 定义学生结构体  struct Student  {      int id;        // 学号      double score;  // 成绩  };    void max(Student\* students, int n)  {      int max = 0;      for (int i = 1; i < n; i++)      {          if (students[i].score > students[max].score)          {              max= i;          }      }        cout << "最高成绩的学生学号是: " << students[max].id << endl;  }    int main()  {      // 创建一个包含5个学生的数组      Student students[5] =      {          {1, 85},          {2, 92},          {3, 78},          {4, 88},          {5, 95}      };        // 调用max函数，查找并输出成绩最高的学生学号      max(students, 5);        return 0;  } | |



5.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | #include <iostream>  using namespace std;  class Point  {  public:        Point(int a, int b)      {          this->x = a;          this->y = b;      }        void setPoint(int i, int j)      {          this->x += i;          this->y += j;      }        void display()      {          cout << "(" << x << "," << y << ")" << endl;      }  private:      int x;      int y;    };  int main()  {      Point p(60, 80);      p.setPoint(20, 30);      p.display();      return 0;  } | |



**总结**

**在本次实验中我初步熟悉了类与对象有关的内容，了解了类的封装，也认识到编程要熟能生巧。**