**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。

【程序设计，算法分析】

1. #include<iostream>

using namespace std;

class Time // 定义Time类

{

private://应实验要求，将数据成员改为私有的

int hour;

int minute;

int sec;

public://因为数据成员为私有，应定义成员函数将其在类内改变

void cintime()//输入要在类内

{

cin >> hour; //输入设定的时间

cin >> minute;

cin >> sec;

}

void showtime()//数据成员的展示也应调用成员函数

{

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

}

};

int main()

{

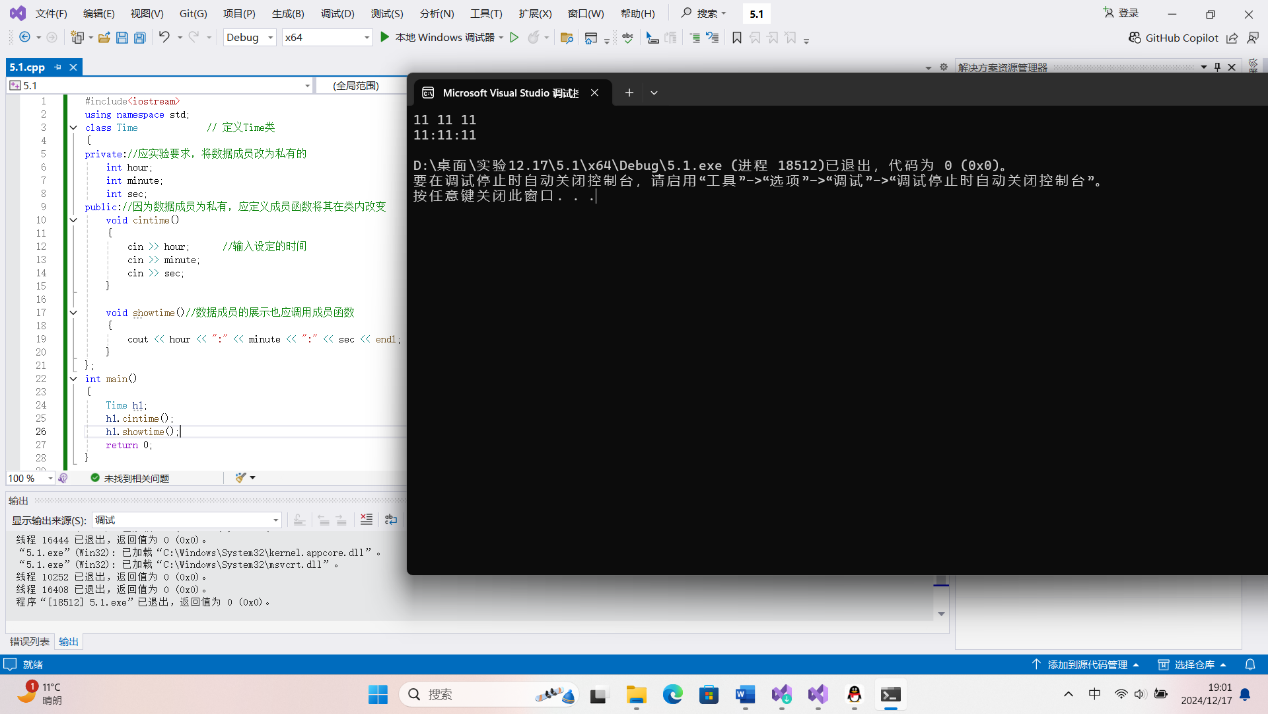
Time h1;

h1.cintime();

h1.showtime();

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()

{

cout << "姓名：";

cin.getline(name, 20);//注意cin.getline函数与cin连用的bug，要提前

cout << "学号:";

cin >> num;

cout << "性别：(m/f)";

cin >> sex;

}

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 << "num：" << num << endl;

cout << "name：" << name << endl;

cout << "sex：" << sex << endl;

}

#include <iostream>//这里是main函数

#include"student.h "

using namespace std;

int main()

{

Student stud;

stud.set\_value();

stud.display();

return 0;

}

3. #include <iostream>

using namespace std;

class cuboid//定义长方体类

{

private:

int length, width, height;//定义长宽高

public:

void set\_cin()

{

cout << "输入长宽高：";

cin>>length ;

cin>>width ;

cin>>height ;

}

void V()

{

int v = length \* width \* height;//计算体积

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

}

};

int main()

{

cuboid c1, c2, c3;

c1.set\_cin();

c2.set\_cin();

c3.set\_cin();

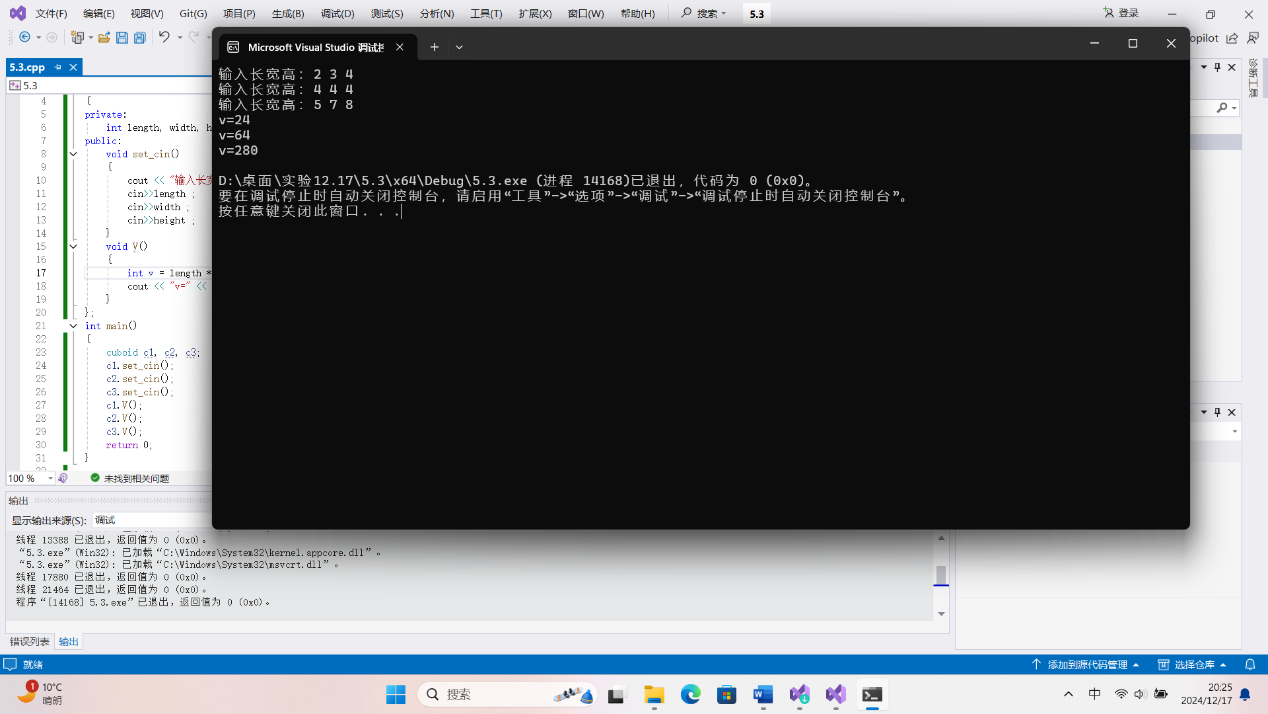
c1.V();

c2.V();

c3.V();

return 0;

}



4. #include <iostream>

using namespace std;

class student

{

friend void max(student\* a, student arr[]);//定义友元函数以解决调用问题

private:

int id, score;//类外难以调用，考虑友元函数

public:

student()

{

id = 0;

score = 0;

}

void set()

{

cout << "学号及分数:";

cin >> id;

cin >> score;

}

void show()

{

cout << id << "号有最高分" << score;

}

};

void max(student\* a, student arr[])//用指向类的指针做参数

{

a = &arr[0];//初始化指针

for (int i = 1; i < 5; i++)

{

if (arr[i].score > a->score)

a = &arr[i];

}

a->show();

}

int main()

{

student stud[5];//定义学生类数组

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

{

stud[i].set();

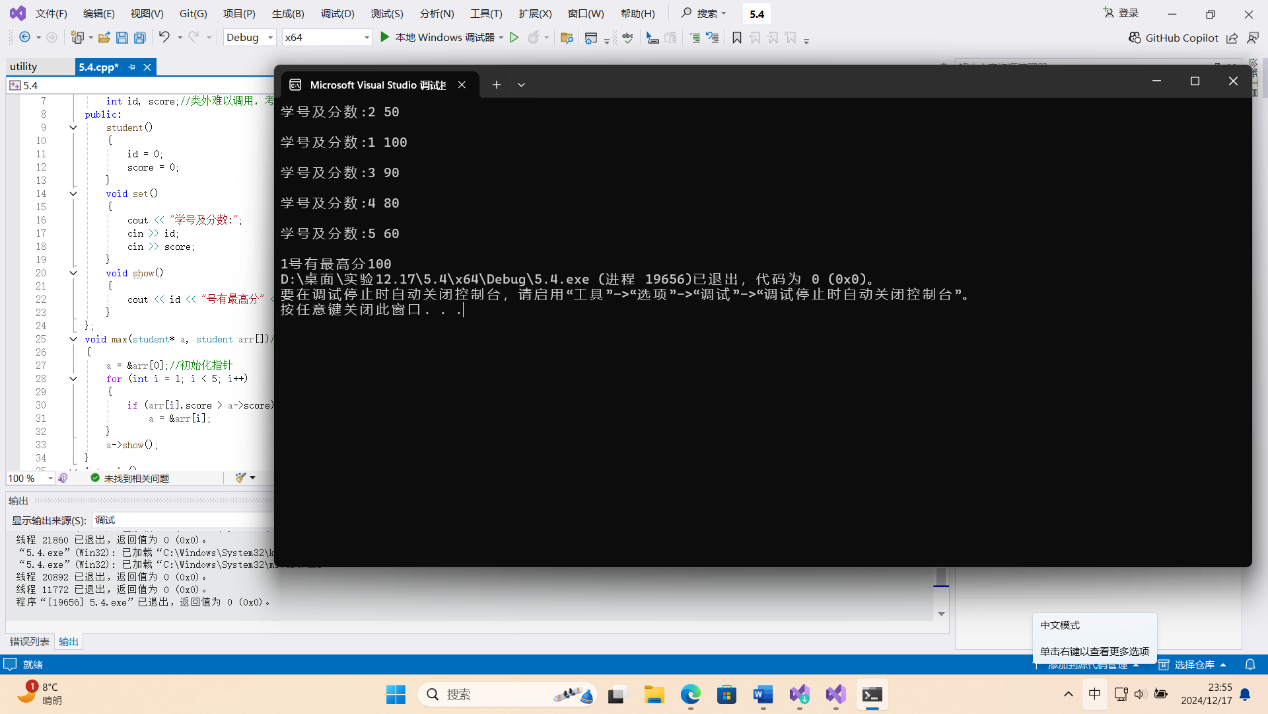
cout << endl;

}

student\* a=nullptr;

max(a, stud); //参数不传数组，首地址即可

return 0;

}

5. #include <iostream>

using namespace std;

class point

{

private:

int x, y;

public:

point()//构造函数，将x、y坐标值初始化为（60,80）

{

x = 60;

y = 80;

}

void setPoint(int i, int j)//公有成员函数void setPoint(int i, int j)将坐标值修改为(60+i,80+j)

{

x += i;

y += j;

}

void display()//公有成员函数display()输出修改后的坐标值

{

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

}

};

int main()

{

int i, j;

point p1;

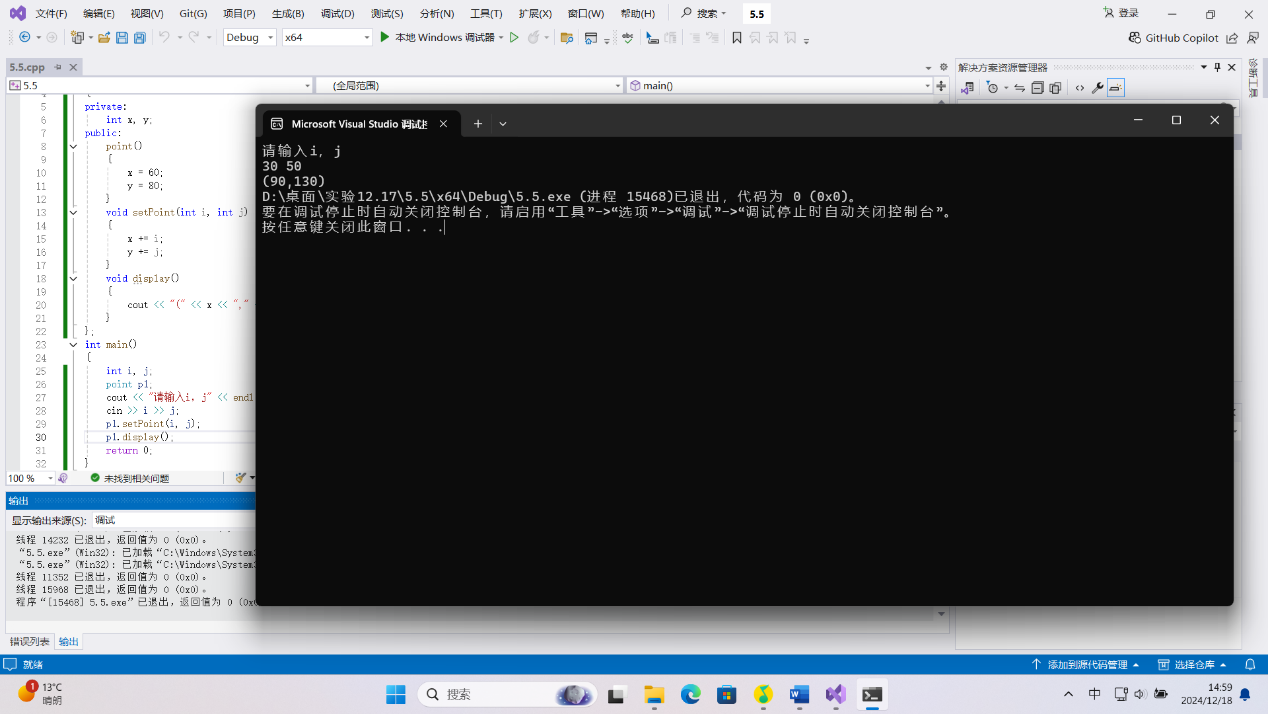
cout << "请输入i，j" << endl;

cin >> i >> j;

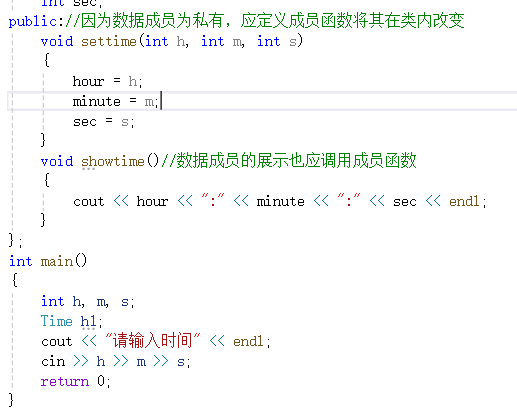
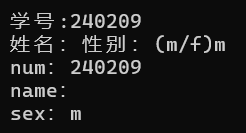
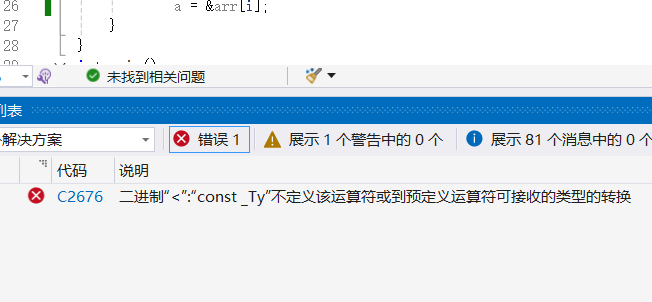
p1.setPoint(i, j);

p1.display();

return 0;

}

【遇到问题及解决方案】

1. 5.1一开始没有仔细审题，没有考虑到输入也要在类内，于是就按照寻常的改变私有数据成员的方法编写。后来发现了问题所在，将其改正了过来。
2. 5.2中一开始按照实验所给顺序并使用了cin.getline函数，结果这一步直接被跳过，如图所示。不知道是什么情况，于是询问学长，发现这是一个cin.getline与cin连用导致的bug，将顺序调换后得到解决。
3. 5.4原本在类外定义max函数时发现无法调用private的数据成员，后来想到了用友元函数，成功调用数据。可后来在测试时出现了大错，最初寻求室友帮助，大伙都说逻辑上没问题，始终找不到错误。最后寻求外援，发现是最低级的参数类型传错，报出了“高级”的错误，这才得以改正。

【体会】

在本次 C++ 类与对象实验课中，我收获颇丰，同时也深刻认识到了自己的不足。

通过实验，我对类与对象的概念有了更直观的理解，学会了如何封装数据和函数，以及如何通过对象来调用类的成员，这使我对面向对象编程的优势有了切实的体会，代码的组织性和逻辑性更加清晰，易于维护和扩展。

然而，实验过程并非一帆风顺。在编写代码时，由于我的粗心大意，在传递函数参数时出现了错误，导致程序运行结果与预期完全不符，花费了大量时间去排查，才发现是如此低级的失误。这让我明白，编程时必须保持高度的专注和严谨，一个小的疏忽可能引发严重的后果。

这次经历也促使我反思自己的学习态度和编程习惯。今后，我会更加认真地对待每一行代码，仔细检查语法和逻辑，养成良好的代码注释习惯，以便在出现问题时能够快速定位和解决。同时，我会加强对基础知识的学习和巩固，提高自己的编程能力，减少因粗心导致的错误，为今后更复杂的编程任务打下坚实的基础。