**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?

1. 改写的程序的代码:

#include<iostream>

using namespace std;

class Time // 定义Time类

{

private:

int hour;

int minute;

int sec;

public:

void In()

{

cin >> hour; //输入设定的时间

cin >> minute;

cin >> sec;

}

void Out()

{

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

}

};

int main()

{

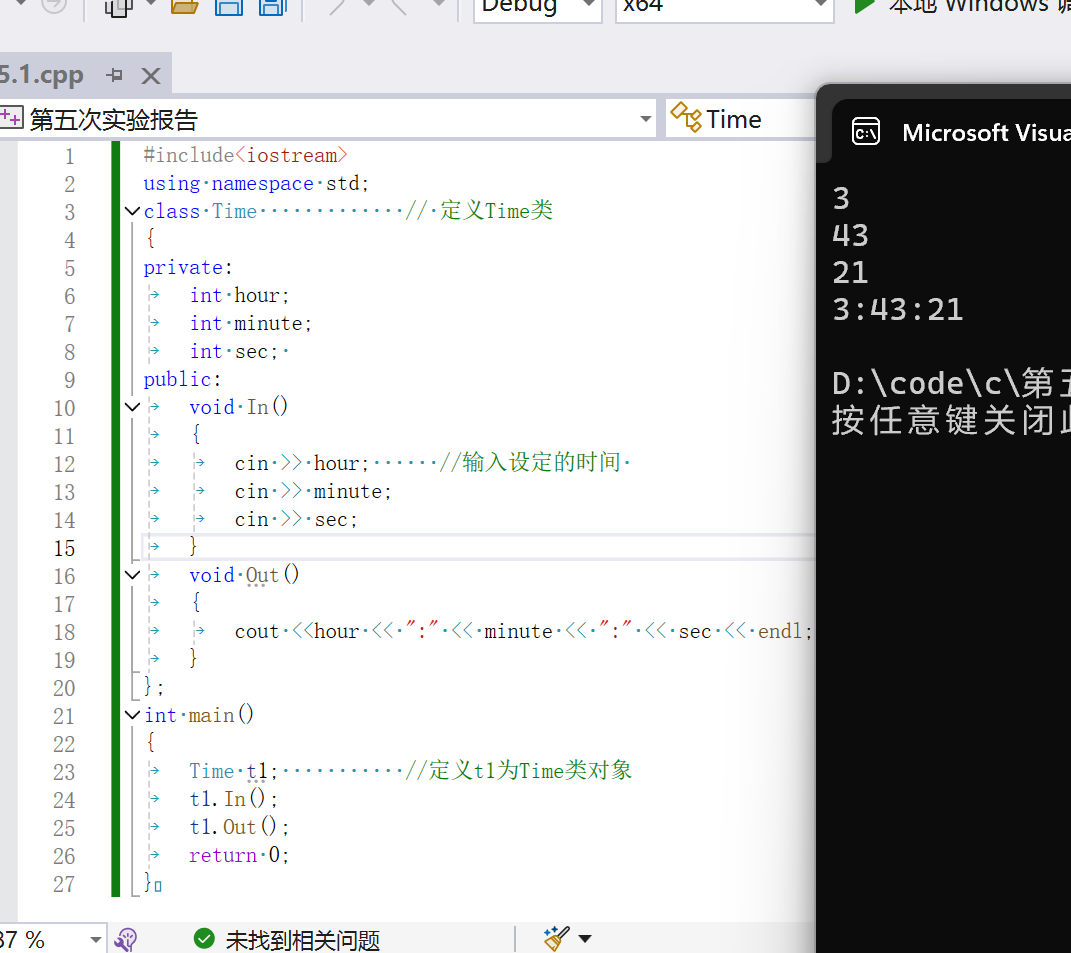
Time t1; //定义t1为Time类对象

t1.In();

t1.Out();

return 0;}

1. 改写程序的运行结果:



1. 回答问题:成员函数In和Out用于输入和输出时间,应该指定为公有的;数据成员应该被设置为私有的(封装对象的状态,保护数据,防止外界直接访问和修改);成员函数最好放在类里面定义,比如与类的内部状态紧密相关的函数(这样可以保存代码的组织性和可读性,方便访问类的私有成员);不需要访问类的私有成员，或者与类的内部状态关系不大的成员函数，可以放在类外定义,比如友元函数。(减少类的复杂性，提高代码的重用性)

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。

1. 完善的程序代码:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include "student.h"

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(int a,const char\*b, char c)

{

num = a;

strcpy(name, b);

sex = c;

}

Student::Student(int n,const char\* nm, char s)

{

num = n;

strcpy(name, nm);

sex = s;

}

Student::Student(): num(0),sex('U')

{

name[0] = '\0';

}

int main()

{

Student stud; //定义对象

stud.set\_value(007, "tcg", 'm');

stud.display(); //执行stud对象的display函数

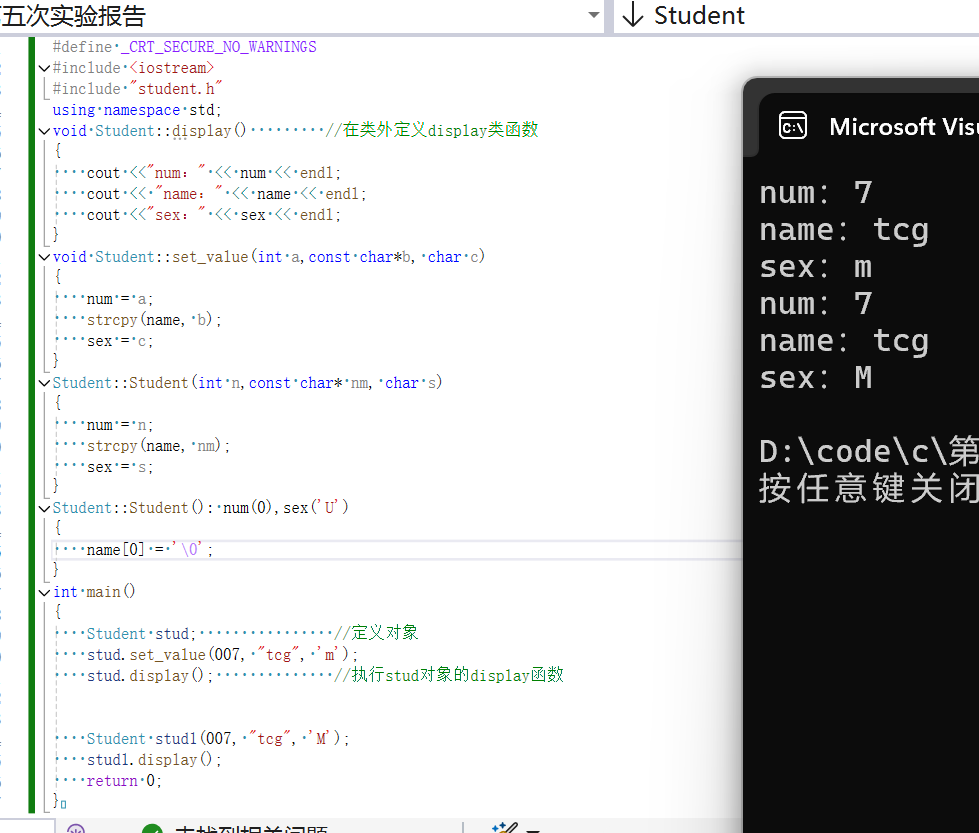
Student stud1(007, "tcg", 'M');

stud1.display();

return 0;

}

1. 程序的运行结果:



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。

1. 编写的程序代码:

#include<iostream>

using namespace std;

class cft

{

private:

int length;

int width;

int height;

int v;

public:

void In()

{

cin >> length;

cin >> width;

cin >> height;

}

void calculate()

{

v = length \* height \* width;

}

void Out()

{

cout <<"体积是:"<<v<< endl;

}

};

int main()

{

cft mem[3];

for (int i = 0; i < 3; i++)

{

mem[i].In();

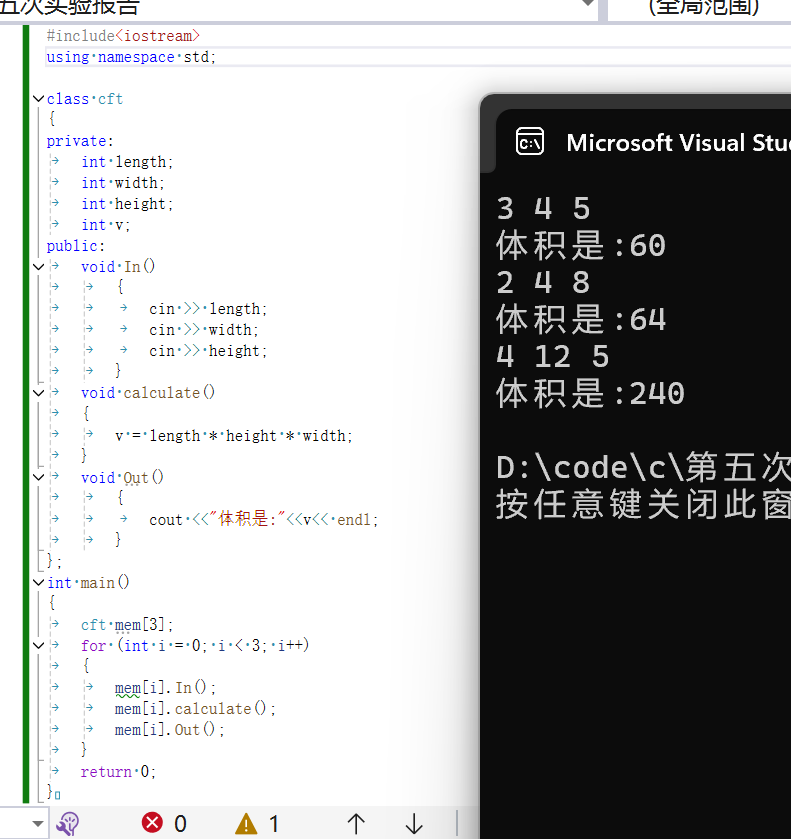
mem[i].calculate();

mem[i].Out();

}

return 0;}

1. 程序的运行结果:



1. 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]编写的程序代码:#include<iostream>

using namespace std;

class Student {

public:

int num;

int score;

void set()

{

int n;

int s;

cin >> n;

cin >> s;

num = n;

score = s;

}

};

void max(Student\* arr,int n)

{

int max = arr[0].score;

int max\_number = arr[0].num;

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

{

if (arr[i].score > max)

{

max = arr[i].score;

max\_number = arr[i].num;

}

}

cout << "成绩最高者的学号是:" << max\_number << endl;

}

int main()

{

Student stus[5];

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

{

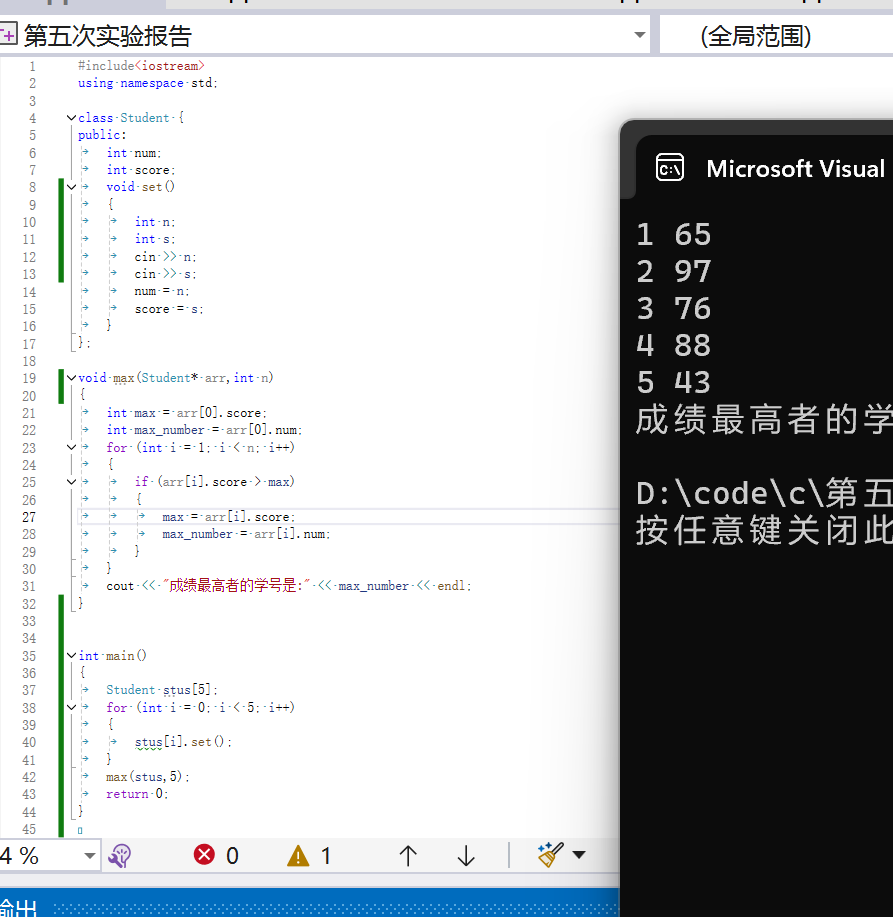
stus[i].set();

}

max(stus,5);

return 0;

}

[2]程序的运行结果:

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。

[1]编写的程序代码:#include<iostream>

using namespace std;

class Point {

private:

int x;

int y;

public:

Point(int xx, int yy)

{

x = xx;

y = yy;

}

void setPoint(int i, int j)

{

x += i;

y += j;

}

void display()

{

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

}

};

int main()

{

Point a(60, 80);

a.display();

a.setPoint(5,6);

a.display();

return 0;

}

[2]程序的运行结果:

