**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。

**实验结果**

1. #include <iostream>

using namespace std;

class Time // 定义Time类

{

private: // 数据成员为公用的

int hour;

int minute;

int sec;

public:

Time(int hour = 0, int minute = 0, int sec = 0)

{

this->hour = hour;

this->minute = minute;

this->sec = sec;

}

void settime(int h, int m, int s)

{

hour = h;

minute = m;

sec = s;

}

void showtime()

{

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

}

};

int main()

{

Time t1; //定义t1为Time类对象

t1.showtime();

int h, m,s;

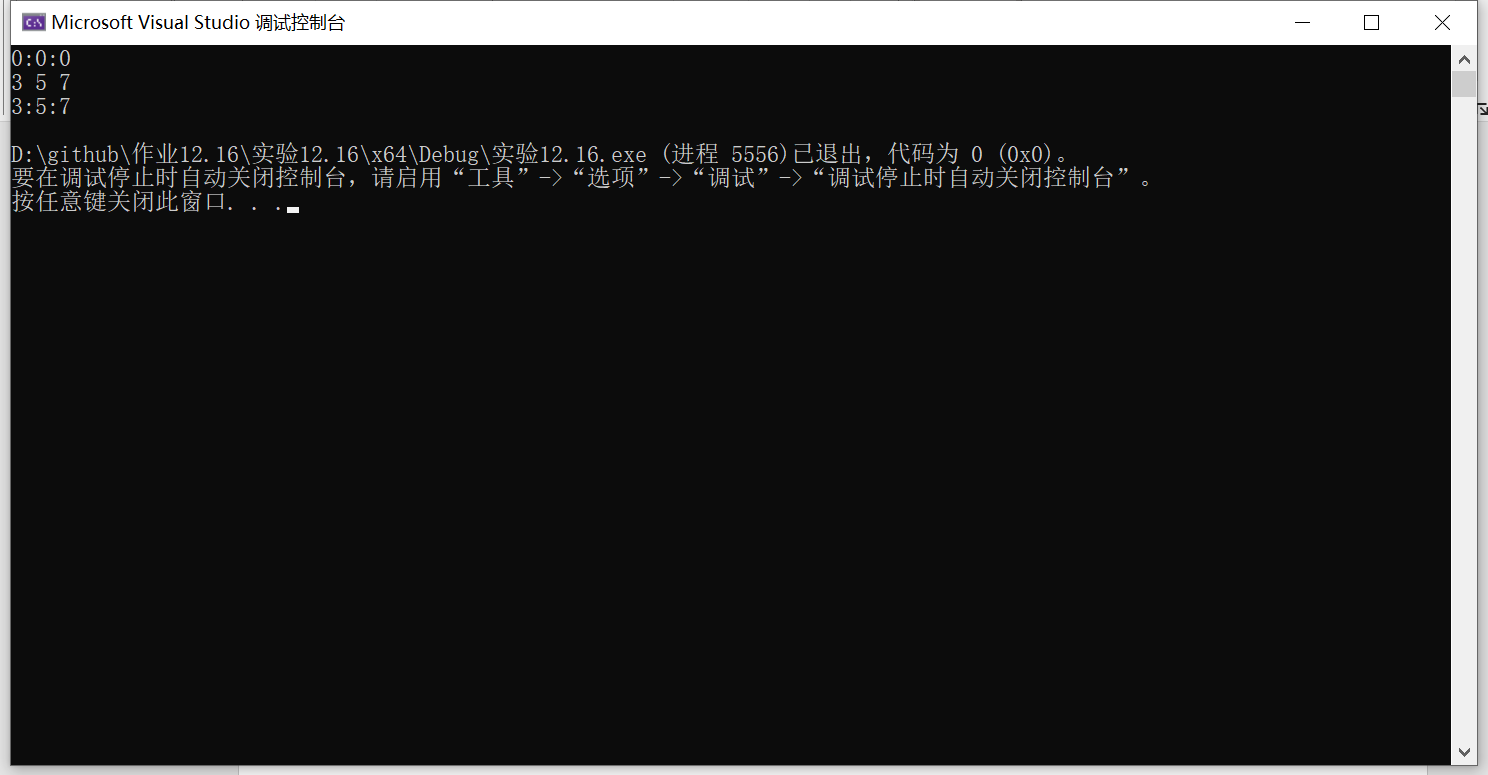
cin >> h >> m >> s;

t1.settime(h, m, s);

t1.showtime();

return 0;

}



变量成员应该私有，函数类成员应该用公有，只需在类内使用的在类内定义，在外面用的多的在类外定义。

2

实验5t2student.h

class Student //类声明

{

private:

int num;

char name[20];

char sex;

public: //公用成员函数原型声明

Student(int num=000, const char name[]="lihua", char sex='n')

{

this->num = num;

for (int i = 0;i < 20;i++)

{

this->name[i] = name[i];

}

this->sex = sex;

}

void display();

void set\_value(int num, const char name[], char sex)

{

this->num = num;

for (int i = 0;i < 20;i++)

{

this->name[i] = name[i];

}

this->sex = sex;

}

};

实验5t2student.cpp

#include <iostream>

#include "实验5t2student.h " //不要漏写此行，否则编译通不过

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;

}

main

#include <iostream> //将类声明头文件包含进来

#include "实验5t2student.h"

int main()

{

Student stud; //定义对象

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

stud.display(); //执行stud对象的display函数

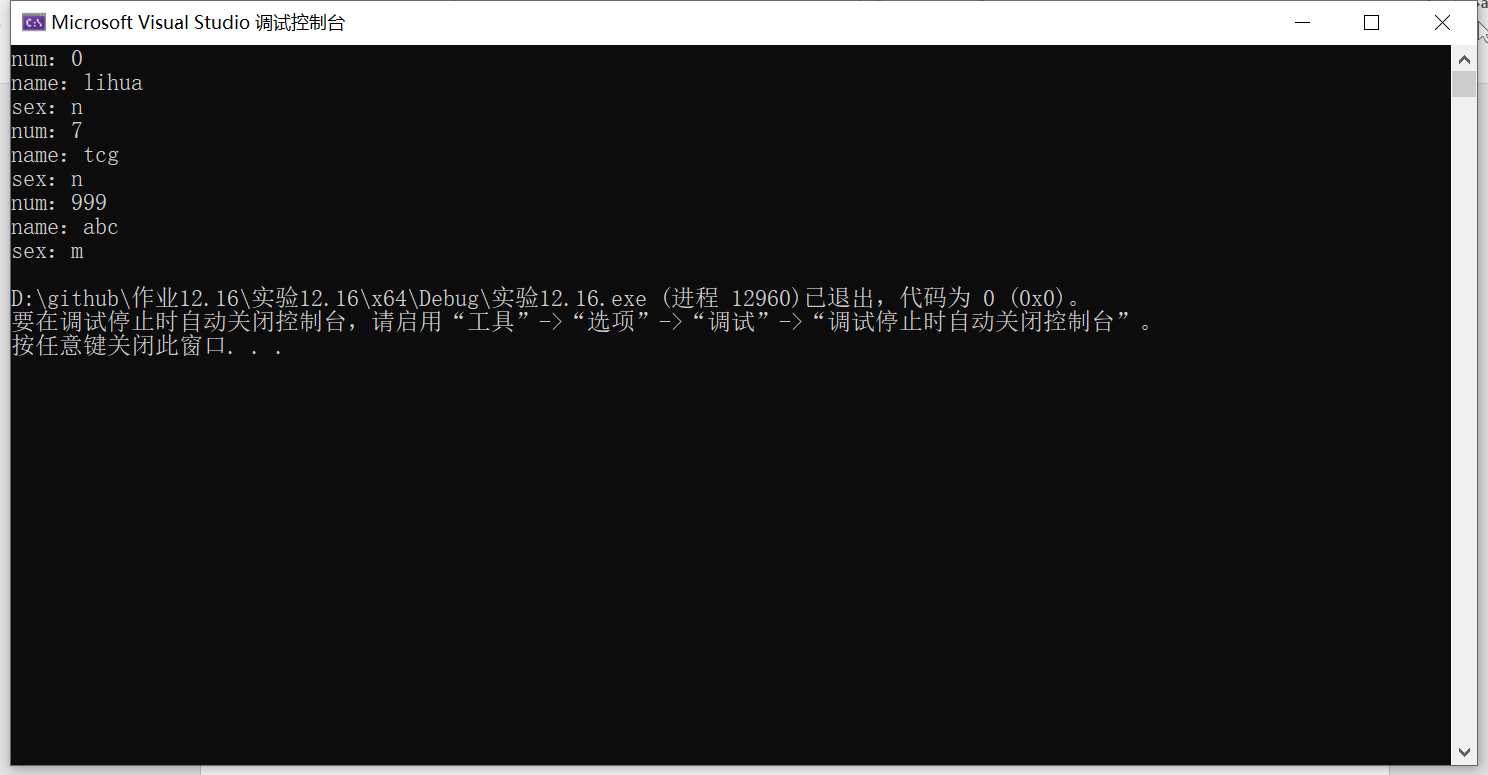
stud1.display();

stud1.set\_value(999, "abc", 'm');

stud1.display();

return 0;

}



3

#include<iostream>

using namespace std;

class Cfz

{

private:

double length;

double width;

double height;

public:

Cfz(double length=0.0, double width=0.0, double height=0.0)

{

this->length = length;

this->width = width;

this->height = height;

}

void set(double length, double width, double height)

{

this->length = length;

this->width = width;

this->height = height;

}

void calculate()

{

cout << length \* width \* height << endl;

}

};

int main()

{

for (int i = 1;i < 4;i++)

{

Cfz ci;

double h, l, w;

cin >> l >> w >> h;

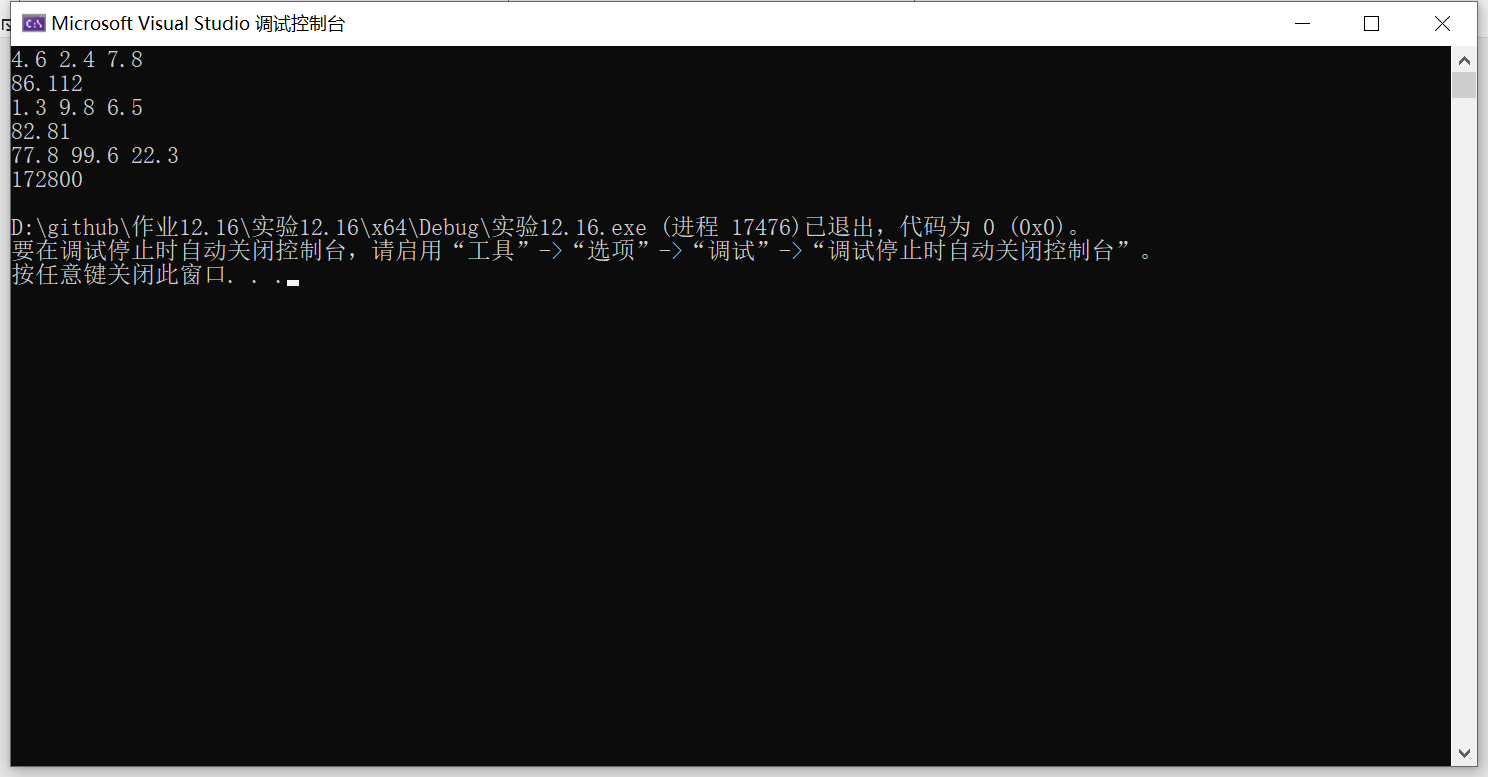
ci.set(l, w, h);

ci.calculate();

}

return 0;

}



4

#include<iostream>

using namespace std;

class Student

{

public:

int grade;

char xh[10];

Student(const char xh[]="0000000000", int grade=0)

{

for (int i = 0;i < 10;i++)

{

this->xh[i] = xh[i];

}

this->grade = grade;

}

};

void max(Student\* sp)

{

char a[10];

int n = 0;

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

{

if (sp[i].grade > n)

{

n = sp[i].grade;

for (int j = 0;j < 10;j++)

{

a[j] = sp[i].xh[j];

}

}

}

for (int i = 0;i < 10;i++)

{

cout << a[i];

}

}

int main()

{

Student st[5] = { Student("0000000001",78),

Student("0000000002",69),

Student("0000000003",95),

Student("0000000004",75),

Student("0000000005",91) };

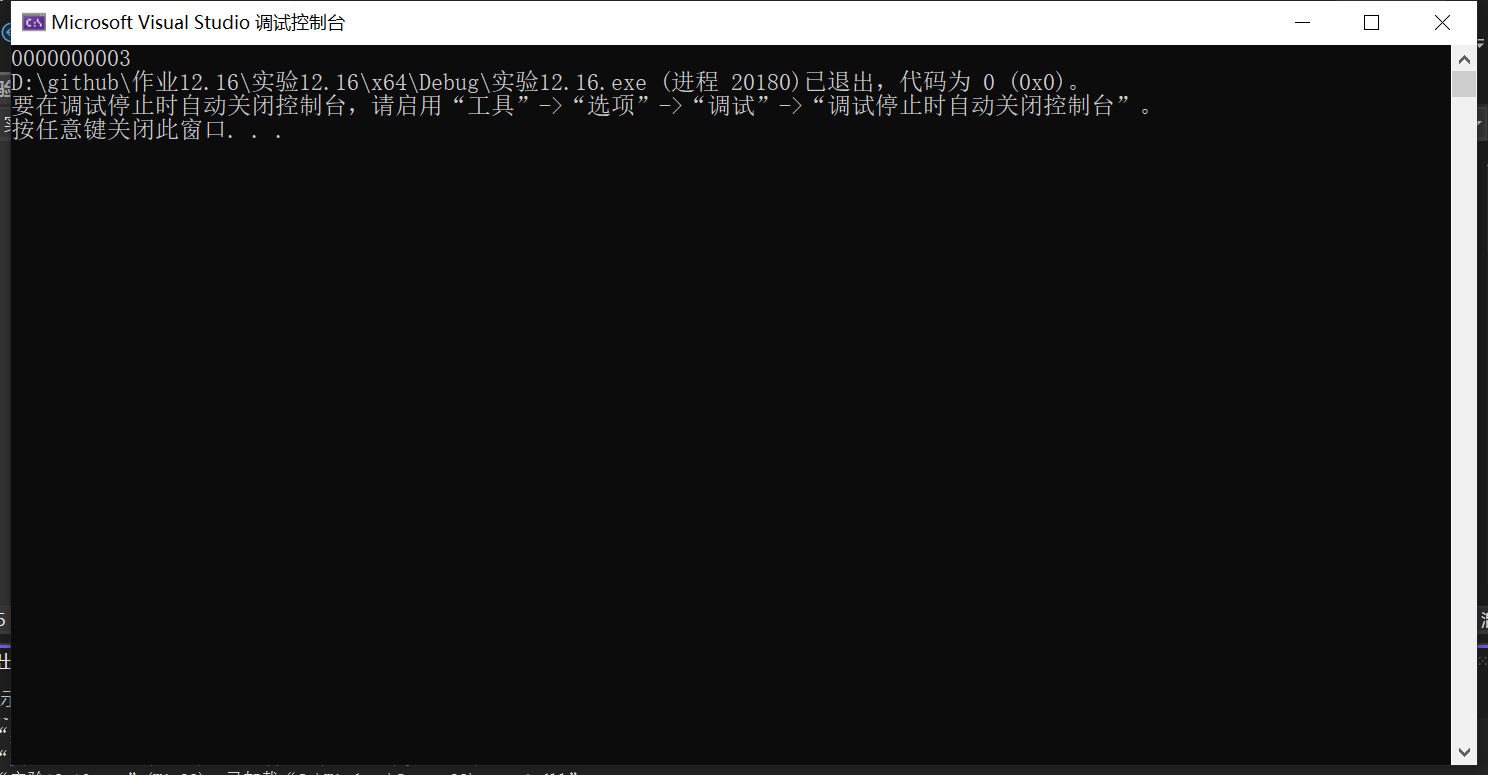
Student\* sp;

sp = st;

max(sp);

return 0;

}



5

#include<iostream>

using namespace std;

class Point

{

private:

int x;

int y;

public:

Point(int x = 0, int y = 0)

{

this->x = x;

this->y = y;

}

void setpoint(int i,int j)

{

x += i;

y += j;

}

void display()

{

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

}

};

int main()

{

Point p(60,80);

p.display();

p.setpoint(15, 25);

p.display();

return 0;

}

