**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?

#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:

void inputTime() {

cin >> hour; //输入设定的时间

cin >> minute;

cin >> sec;

}

void outputTime(){

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

}

};

int main()

{

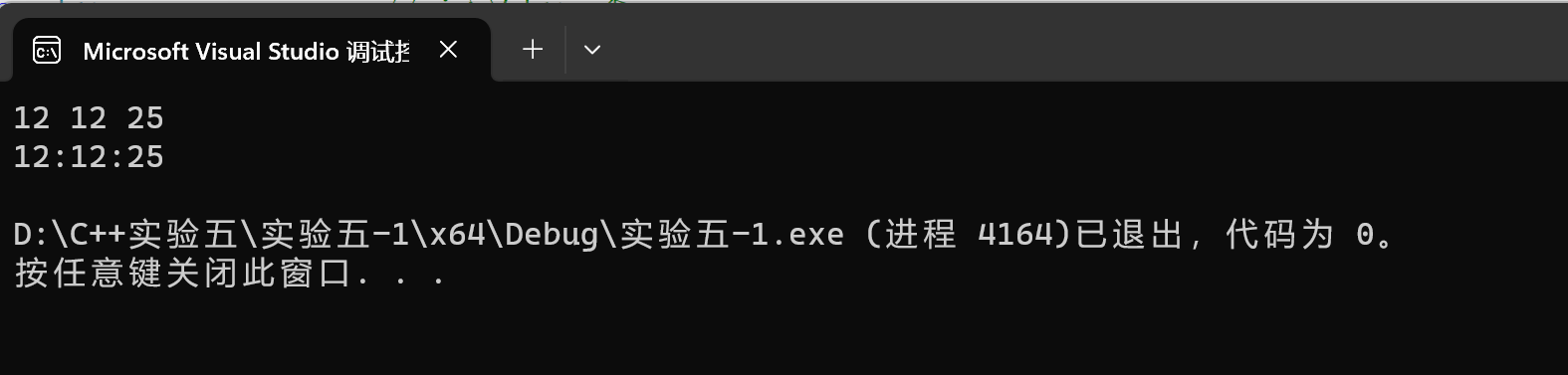
Time t1; //定义t1为Time类对象

t1.inputTime();

t1.outputTime();

return 0;

}



答：1.那些需要被外部代码调用的成员函数

2.防止外部直接访问和修改的成员变量

3.需要访问类的私有成员，或函数涉及到操作类的对象，修改对象的状态或使用对象的数据成员

4.与类无直接关系的独立功能的函数，或者友元函数。

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。

//student.h

#pragma once

#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(int n, string nm, char s);

private:

int num;

string name;

char sex;

};

//student.cpp

#include<iostream>

#include<string>

#include"student.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;

}

void Student::set\_value(int n, string nm, char s)

{

num = n;

name = nm;

sex = s;

}

//main.cpp

#include <iostream>

#include "student.h" //将类声明头文件包含进来

using namespace std;

int main()

{

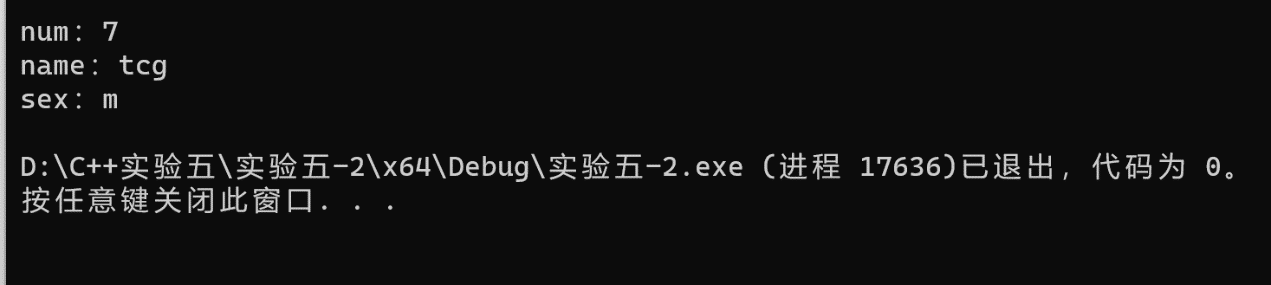
Student stud; //定义对象

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

stud.display(); //执行stud对象的display函数

return 0;

}



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。

#include<iostream>

using namespace std;

class Cuboid {

private:

double length;

double width;

double height;

public:

void inputDimensions() {

cout << "请输入长方柱的长、宽、高：" << endl;

cin >> length >> width >> height;

}

double calculateVolume() {

return length \* width \* height;

}

void outputVolume() {

cout << "长方柱的体积为：" << calculateVolume() << endl;

}

};

int main()

{

Cuboid cuboid1, cuboid2, cuboid3;

cuboid1.inputDimensions();

cuboid2.inputDimensions();

cuboid3.inputDimensions();

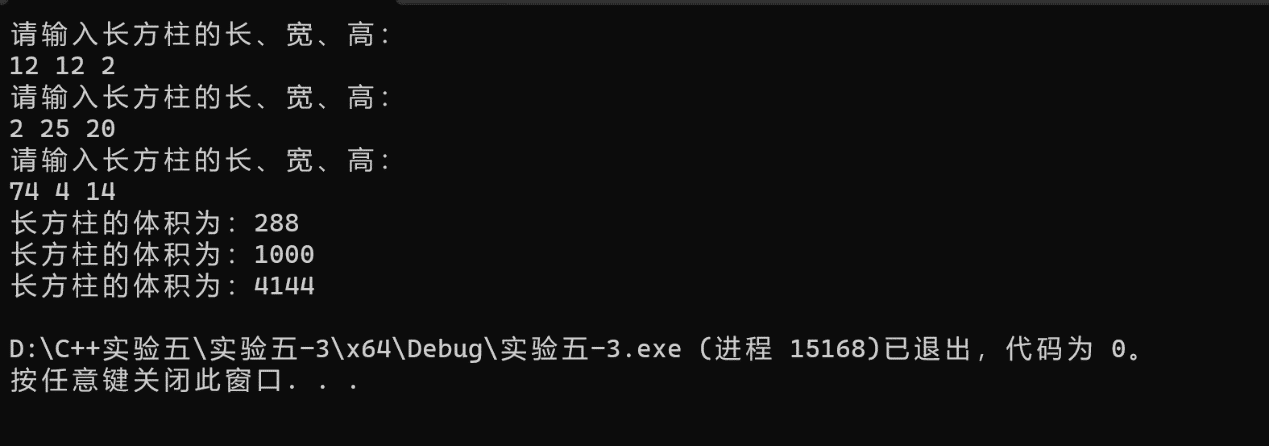
cuboid1.outputVolume();

cuboid2.outputVolume();

cuboid3.outputVolume();

return 0;

}



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。

#include<iostream>

using namespace std;

class Students {

private:

int studentID;

double score;

public:

Students(int id, double s) {

studentID = id;

score = s;

}

int getStudentID() {

return studentID;

}

double getScore() {

return score;

}

};

void max(Students\* students[], int size) {

int maxIndex = 0;

double maxScore = students[0]->getScore();

for (int i = 1; i < size; i++) {

if (students[i]->getScore() > maxScore) {

maxScore = students[i]->getScore();

maxIndex = i;

}

}

cout << "最高成绩者的学号是：" << students[maxIndex]->getStudentID() << endl;

}

int main() {

Students\* students[5];

students[0] = new Students(1, 85.5);

students[1] = new Students(2, 76.0);

students[2] = new Students(3, 92.3);

students[3] = new Students(4, 88.9);

students[4] = new Students(5, 78.6);

max(students, 5);

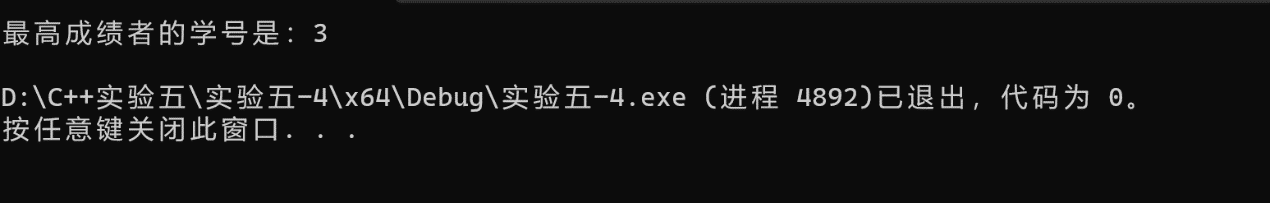
for (int i = 0; i < 5; i++) {

delete students[i];

}

return 0;

}



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。

#include<iostream>

using namespace std;

class Point {

private:

int x;

int y;

public:

Point(){

x = 60;

y = 80;

}

void setPoint(int i, int j) {

x = 60 + i;

y = 80 + j;

}

void display() {

cout << "坐标值：（" << x << "," << y << ")" << endl;

}

};

int main() {

Point p1;

cout << "初始坐标值：" << endl;

p1.display();

p1.setPoint(1, 2);

cout << "修改后的坐标值为：" << endl;

p1.display();

return 0;

}

