**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.**

#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() :hour(0), minute(0), sec(0) {}// 构造函数，默认构造函数

void setTime(int h, int m, int s) // 设置时间的函数

{

hour = h;

minute = m;

sec = s;

}

void printTime() const {

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

}

};// 打印时间的函数

int main()

{

Time t1;

int h, m, s;//定义t1为Time类对象

cout << "Enter time(hour minut sec):";

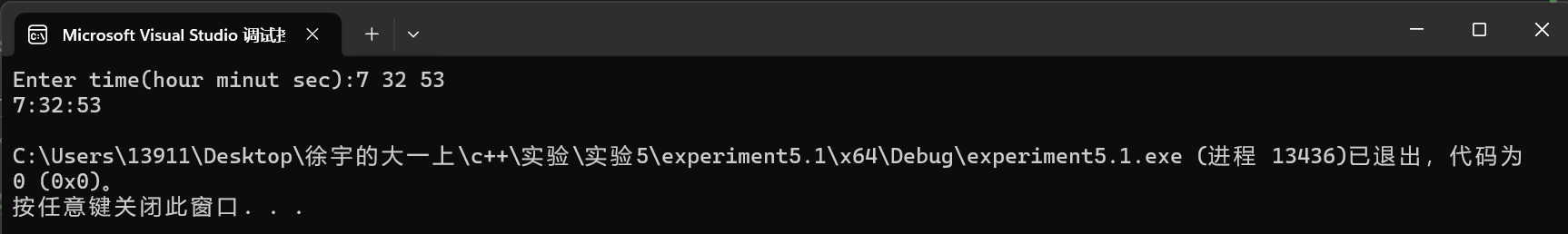
cin >> h >> m >> s;

t1.setTime(h, m, s); // 设置时间

t1.printTime(); // 打印时间

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() //赋值成员函数

{

cout << "set name is:";

cin.getline(name, 20);//使用getline输入

cout << "set sex is:(m/f) ";

cin >> sex;

cout << "set num is:";

cin >> num;

}

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 << "name:" << name << endl;

cout << "num:" << num << endl;

cout << "sex:" << sex << endl;

}

//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

stud.set\_value(); //执行stud对象的set\_value函数

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.**

#include <iostream>

using namespace std;

class Cuboid{

private:

double length, width, height;

public:

void input(){

cout << "Enter the length, width, height: ";

cin >> length >> width >> height;

}

double calculatevolume(){

return length \* width \* height;

}

void displayVolume(){

cout << "volume: " << calculatevolume() << endl;

}

};

int main(){

Cuboid cuboid;

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

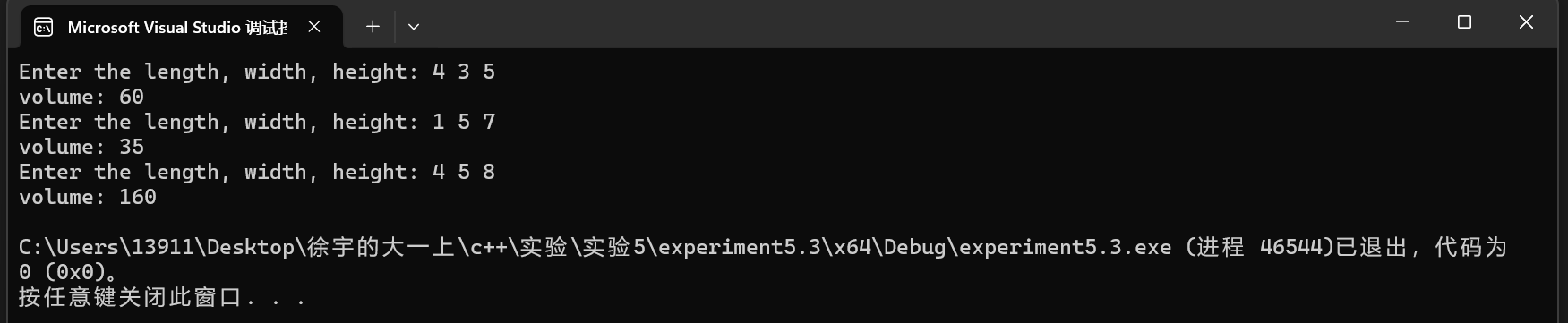
cuboid.input();

cuboid.displayVolume();

}

return 0;

}

****

**4.**

#include <iostream>

using namespace std;

class Student {

private:

int id;

double score;

public:

Student(int i, double s) :id(i), score(s) {}

double getScore() {return score;}

int getId() {return id;}

};

double max(Student students[], int size) {

double highest=students[0].getScore();

int id=students[0].getId();

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

if (students[i].getScore() > highest) {

highest = students[i].getScore();

id= students[i].getId();

}

}

cout<<"The highest score is "<<highest<<" and the student's id is "<<id<<endl;

return highest;

}

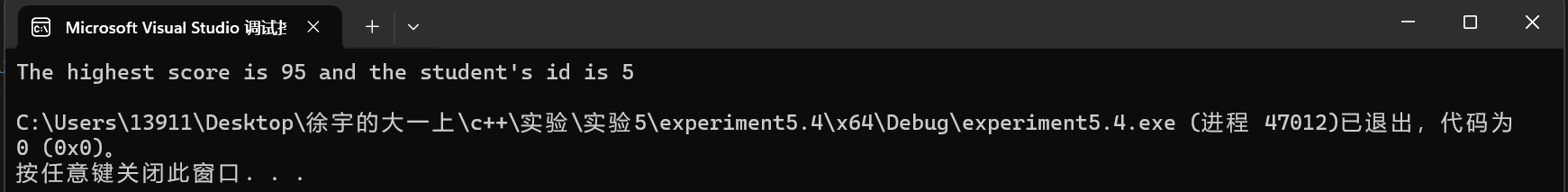
int main() {

Student students[5] = {Student(1, 85), Student(2, 92), Student(3, 78),Student(4,89),Student(5,95)};

max(students, 5);

return 0;

}

****

**5.**

#include <iostream>

using namespace std;

class Point {

private:

int x, y;

public:

Point(int ix=60, int iy=80):x(ix), y(iy) {}

void setPoint(int i, int j) {

x = i + 60;

y = i + 80;

}

void display() {

cout << "The Point is:(" << x << "," << y <<")"<< endl;

}

};

int main() {

Point pt;

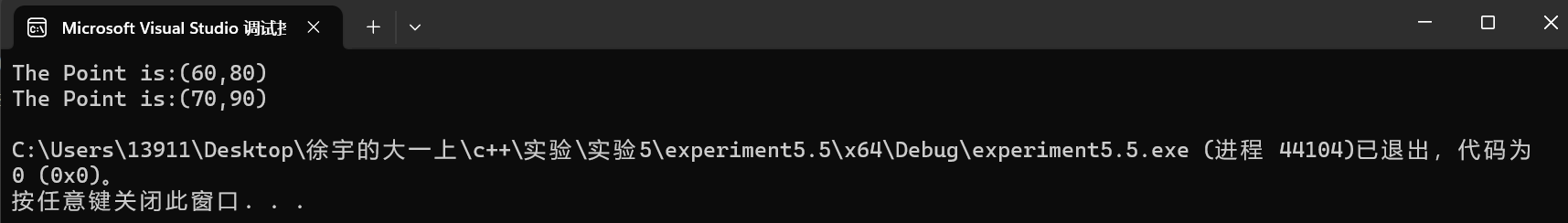
pt.display();

pt.setPoint(10, 20);

pt.display();

return 0;

}

****

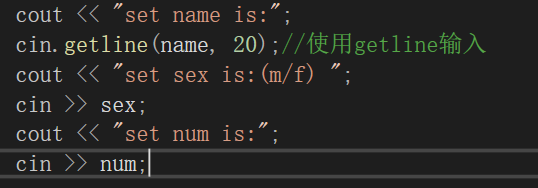
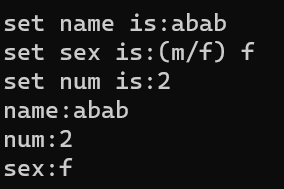
**【遇到的问题与解决方法】**

**1.cin.getline()无法正常执行**

在进行实验5.2的过程中，需要用到cin.getline()函数，但是在使用该函数程，程序运行后，发现这一步被跳了过去。

使用f11分析，发现在进行这一步的时候无法进行输入，再次按下f11，这一步被直接跳了过去。

首先考虑自己是不是程序语法出现了问题，但是编译器并没有报错，且在检查之后，基本语法没有出现错误。

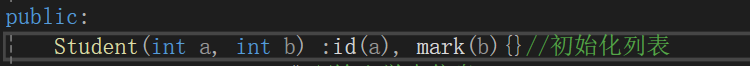
再考虑是不是因为将cin.getline()与cin连用，导致出现问题。在调整cin.getline()与cin的顺序后，发现程序可以正常运行，问题解决。

**2.初始化时忘了构造函数**

在实验5.4中，定义了student类的数组，在尝试对该数组进行赋值时，编译器报错。

在回想为何会报错时，才想起来自己把类当做了结构体，没有构造函数就赋值定然会报错。

遂构造函数。



**【体会】**

通过本次 C++ 类与对象的上机实验，我收获颇丰，对面向对象编程有了更深刻的理解和认识。

在实验过程中，我逐步掌握了类的定义和对象的创建与使用方法。深刻体会到类作为一种用户自定义的数据类型，将数据和对数据的操作封装在一起，大大提高了代码的模块化程度和可维护性。例如，在创建一个简单的“学生”类时，将学生的姓名、年龄、成绩等属性以及获取成绩、显示信息等行为封装在类中，使得代码结构清晰，易于理解和修改。

同时，我也学习到了构造函数和析构函数的重要性和作用。构造函数用于初始化对象的数据成员，确保对象在创建时处于一个合理的初始状态；析构函数则用于释放对象占用的资源，避免内存泄漏等问题。正确地使用构造函数和析构函数是编写健壮 C++ 程序的关键环节之一。

然而，实验过程中也并非一帆风顺。在类的继承方面，我遇到了一些困难。理解父类和子类之间的关系等概念需要花费一定的时间和精力。但通过查阅资料和反复调试代码，我最终克服了这些问题，也进一步加深了对这些知识的理解。

这次实验让我认识到，C++ 类与对象的编程思想虽然具有一定的复杂性，但只要掌握了其核心概念和方法，并通过大量的实践练习，就能灵活运用它来解决实际问题，开发出高质量的程序。在今后的学习中，我将继续加强对 C++ 编程的学习和实践，不断提高自己的编程能力。