**计算机程序设计基础（C++）**

**实验报告**

专业班级\_\_\_\_\_\_\_\_软件工程2206班\_\_\_\_\_\_\_

学 号\_\_\_\_\_\_\_\_\_\_\_8209220605\_\_\_\_\_\_\_\_\_

姓 名\_\_\_\_\_\_\_\_\_\_\_\_\_王一丁\_\_\_\_\_\_\_\_\_\_\_\_

**实验报告成绩：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验** | **实验二** | **实验三** | **实验四** | **实验五** | **总评** |
| **成绩** |  |  |  |  |  |

**批阅教师：**

**实验五 类与对象**

1. **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。

1. **实验内容**

1.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

{

public: // 数据成员为公用的

int hour;

int minute;

int sec;

};

int main()

{

Time t1; //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.hour << ":" << t1.minute << ":" << t1.sec << endl;

return 0;

}

2.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(代码已用横线分割)

//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;

};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//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;

}

---------——————————————————————————————

//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(7, "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。

4、编写程序：定义抽象基类 Shape，

由它派生出五个派生类：Circle（圆形）、Square（正方形）、Rectangle（ 长方形）、Trapezoid （梯形）和 Triangle （三角形），用虚函数分别计算各种图形的面积，并求出它们的和。

要求用基类指针数组。使它的每一个元素指向一个派生类的对象。

注：主函数中定义如下对象

Circle circle(10.2);

Square square(3);

Rectangle rectangle(3,4);

Trapezoid trapezoid(2.0,4.5,3);

Triangle triangle(4,5,6);

5、设计一个函数，返回两个向量的交集，函数头如下：

template vector intersection(const vector& v1, const vector& v2)

两个向量的交际为它们的共同包含的元素，编写一个测试程序，提示用户输入两个向量，每个包含 5 个字符串，打印输出它们的交集。

**三、算法分析、程序与结果**

**1.程序：**

**(1) 将数据成员改为私有的**

#include<iostream>

using namespace std;

int main()

{

const int size = 10;

int list[size];

cout << "Enter ten numbers:" << endl;

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

cin >> list[i];

cout << list[0] << " ";

for (int m = 1; m < size; m++)

{

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

{

if (list[m] != list[j])

{

if (j == m - 1)

cout << list[m] << " ";

else

continue;

}

else

break;

}

}

cout << endl;

return 0;

}

**(2) 将输入和输出的功能改为由成员函数实现**

#include <iostream>

using namespace std;

class Time

{

public:

void set\_time(void);

void show\_time(void);

private:

int hour;

int minute;

int sec;

};

Time t;

int main()

{

t.set\_time();

t.show\_time();

return 0;

}

void Time::set\_time(void)

{

cin >> hour;

cin >> minute;

cin >> sec;

}

void Time::show\_time(void)

{

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

}

**(3) 在类体内定义成员函数**

#include <iostream>

using namespace std;

class Time

{

public:

void set\_time(void)

{

cin >> hour;

cin >> minute;

cin >> sec;

}

void show\_time(void)

{

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

}

private:

int hour;

int minute;

int sec;

};

Time t;

int main()

{

t.set\_time();

t.show\_time();

return 0;

}

**2.程序：**

**Student.h**

#pragma once

class Student //类声明

{

public:

void display(); //公用成员函数原型声明

void set\_value();

private:

int num;

char name[20];

char sex;

};

**Student.cpp**

#include<iostream>

using namespace std;

#include"student.h"

void Student::display()

{

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

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

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

}

void Student::set\_value()

{

cin >> num;

cin >> name;

cin >> sex;

}

**Main.cpp**

//main. cpp 主函数模块

#include <iostream>

using namespace std;

#include "student.h" //将类声明头文件包含进来

int main()

{

Student stud;

//定义对象

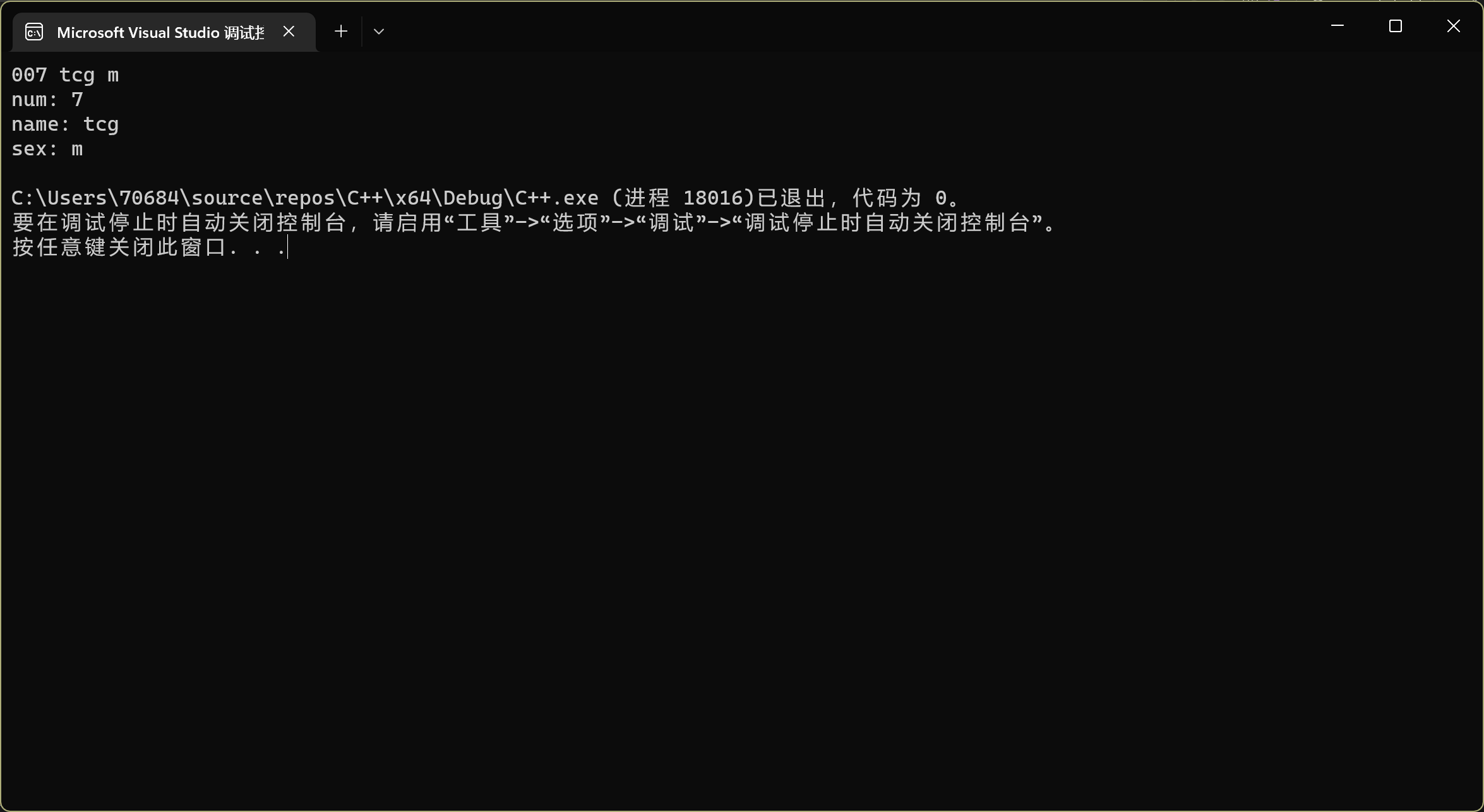
stud.set\_value();

stud.display(); //执行stud对象的 display函数

return 0;

}

**结果：**



**3.程序：**

#include<iostream>

using namespace std;

class Box

{

public:

void get\_value();

float volume();

void display();

private:

float length;

float width;

float height;

};

void Box::get\_value()

{

cout << "please input length,width,height:";

cin >> length >> width >> height;

}

float Box::volume()

{

return (length \* width \* height);

}

void Box::display()

{

cout << volume() << endl;

}

int main()

{

Box box1, box2, box3;

box1.get\_value();

cout << "volume of box1 is ";

box1.display();

box2.get\_value();

cout << "volume of box2 is ";

box2.display();

box3.get\_value();

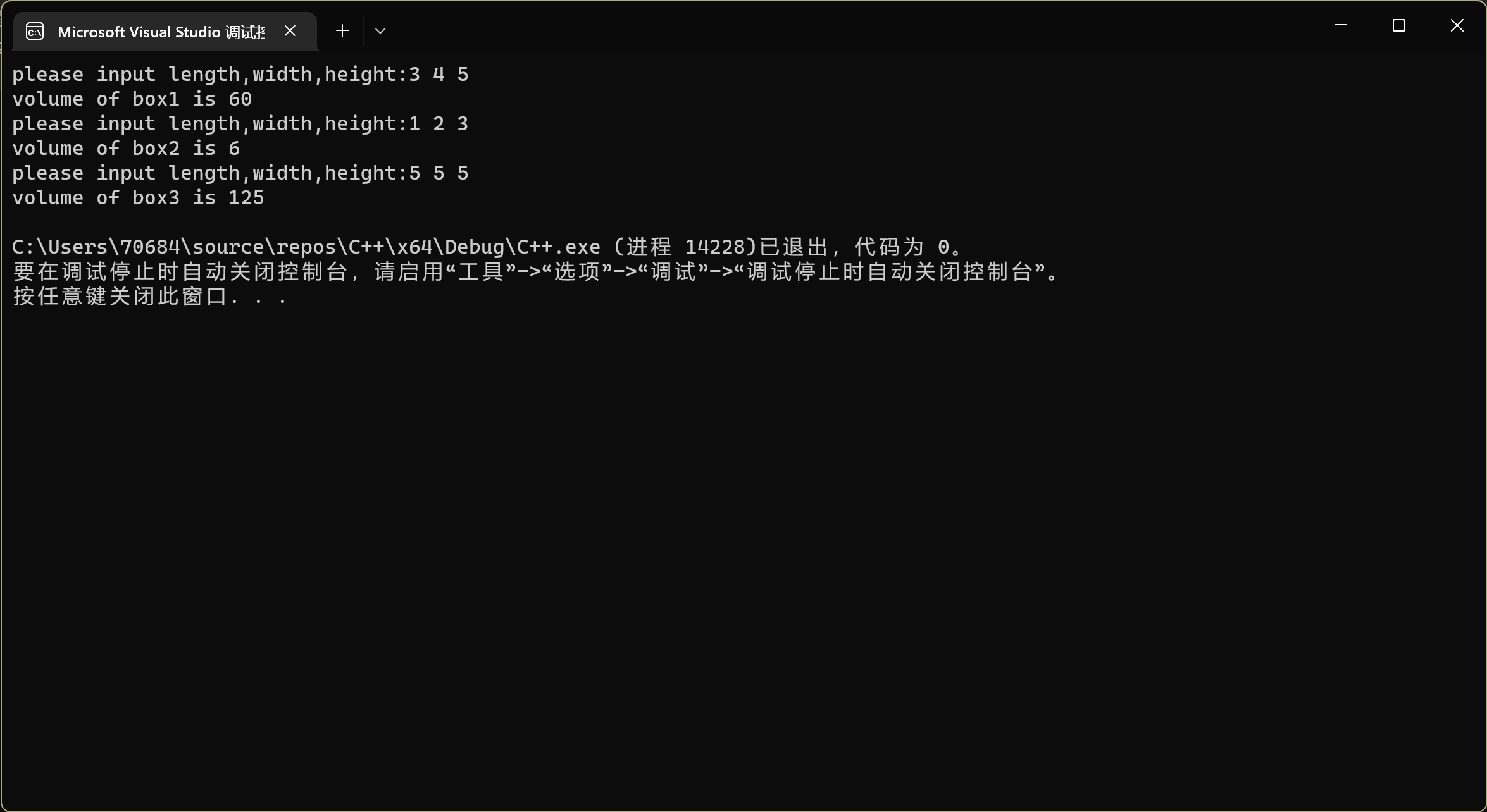
cout << "volume of box3 is ";

box3.display();

return 0;

}

**结果：**

****

**4.程序：**

#include<iostream>

using namespace std;

class Shape

{

public:

virtual double area() const = 0;

};

class Circle :public Shape

{

public:

Circle(double r) :radius(r)

{

}

virtual double area() const

{

return 3.14159 \* radius \* radius;

}

protected:

double radius;

};

class Square :public Shape

{

public:

Square(double s) :side(s)

{

}

virtual double area() const

{

return side \* side;

}

protected:

double side;

};

class Rectangle :public Shape

{

public:

Rectangle(double w, double h) :width(w), height(h)

{

}

virtual double area() const

{

return width \* height;

}

protected:

double width, height;

};

class Trapezoid :public Shape

{

public:

Trapezoid(double t, double b, double h) :top(t), bottom(b), height(h)

{

}

virtual double area() const

{

return 0.5 \* (top + bottom) \* height;

}

protected:

double top, bottom, height;

};

class Triangle :public Shape

{

public:

Triangle(double a, double b, double c) :length1(a), length2(b), length3(c)

{

}

virtual double area() const

{

double p = (length1 + length2 + length3) / 2;

return sqrt(p \* (p - length1) \* (p - length2) \* (p - length3));

}

protected:

double length1, length2, length3;

};

int main()

{

Circle circle(10.2);

Square square(3);

Rectangle rectangle(3,4);

Trapezoid trapezoid(2.0, 4.5, 3);

Triangle triangle(4, 5, 6);

Shape\* pt[5] =

{

&circle,&square,&rectangle,&trapezoid,&triangle

};

double areas = 0.0;

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

{

areas = areas + pt[i]->area();

}

cout << "total of all areas=" << areas << endl;

return 0;

}

**结果：**

****

**5.程序：**

#include "iostream"

#include "vector"

#include "algorithm" //sort函数、交并补函数

#include "iterator" //求交并补使用到的迭代器

using namespace std;

//打印容器vector

void print\_vector(vector<int> v) {

if (v.size() > 0) {

cout << "{";

for (int i = 0; i<int(v.size()); i++) {

cout << v[i] << ",";

}

cout << "\b}";

}

else {

cout << "{}";

}

}

//两个vector求交集

vector<int> vectors\_intersection(vector<int> v1, vector<int> v2) {

vector<int> v;

sort(v1.begin(), v1.end());

sort(v2.begin(), v2.end());

set\_intersection(v1.begin(), v1.end(), v2.begin(), v2.end(), back\_inserter(v));//求交集

return v;

}

int main()

{

vector<int> v1, v2, v;

int a, b, c, d, e, f, g, h, i, j;

cin >> a >> b >> c >> d >> e >> f >> g >> h >> i >> j;

v1.push\_back(a); v1.push\_back(b); v1.push\_back(c); v1.push\_back(d); v1.push\_back(e);

v2.push\_back(f); v2.push\_back(g); v2.push\_back(h); v2.push\_back(i); v2.push\_back(j);

cout << "求v1与v2的交集：";

v = vectors\_intersection(v1, v2);

print\_vector(v);

cout << endl;

return 0;

}**结果：**

****

**四、遇到的问题与解决方法**

**第四题的三角形用到了之前自学的海伦公式，不然只给三条边我不知道该怎么求三角形面积的。第二题也是复习，前面的实验做过头文件，定义cpp和主函数cpp相关的题目，相当于巩固复习了吧。**

**五、体会**

**自从放假以来都没写过代码，这次赶在ddl的前一天完成了实验五，明显感到自己生疏了许多。寒假也要花时间时间来好好学习，不然上学期学习的知识就全忘记了。**