**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。

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。

**5.1**

实验代码

#include <iostream>

using namespace std;

#include<iostream>

using namespace std;

class Time // 定义 Time 类

{

private:

int hour;

int minute;

int sec;

public:

int time()

{

hour = 0;

minute = 0;

sec = 0;

}

int time(int newHour, int newMinute, int newSec)

{

hour = newHour;

minute = newMinute;

sec = newSec;

}

void setHour()

{

cout << "请输入小时: " ;

int newHour;

cin >> newHour;

hour = newHour;

}

void setMinute()

{

cout << "请输入分钟: " ;

int newMinute;

cin >> newMinute;

minute = newMinute;

}

void setSec()

{

cout << "请输入秒: " ;

int newSec;

cin >> newSec;

sec = newSec;

}

void getHour()

{

cout << hour <<"时" << endl;

}

void getMinute()

{

cout << minute <<"分" << endl;

}

void getSec()

{

cout << sec <<"秒" << endl;

}

};

int main()

{

Time t1; //定义 t1 为 Time 类对象

t1.setHour(); //输入设定的时间

t1.setMinute();

t1.setSec();

t1.getHour();

t1.getMinute();

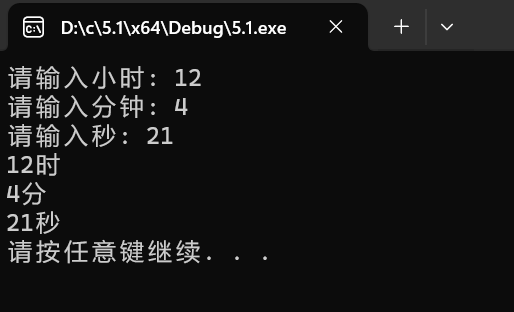
t1.getSec();

system("pause");

return 0;

}

结果截屏



**5.2**

实验代码

student.h

#pragma once

class Student //类声明

{

public: //公用成员函数原型声明

void display();

void set\_value(int \_num, std::string \_name, char \_sex)

{

num = \_num;

name = \_name;

sex = \_sex;

}

private:

int num;

std::string name;

char sex;

};

student.cpp

#include <iostream>

#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;

}

main.cpp

#include <iostream> //将类声明头文件包含进来

#include "student.h"

using namespace std;

int main()

{

Student stud; //定义对象

//Student stud1(007, "tcg", 'm');

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

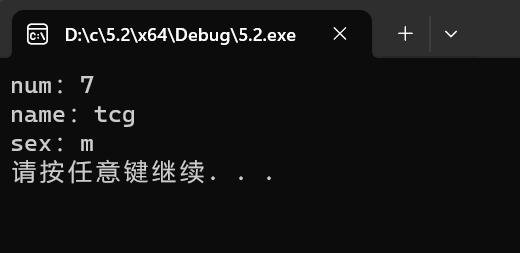
stud.display(); //执行stud对象的display函数

system("pause");

return 0;

}

结果截屏



**5.3**

实验代码

Rectangle.h

#pragma once

class Rectangle

{

public:

Rectangle()

{

length = 1;

width = 1;

heigh = 1;

}

Rectangle(double newLength, double newWidth, double newHeigh)

{

length = newLength;

width = newWidth;

heigh = newHeigh;

}

double length;

double width;

double heigh;

double getVolume()

{

return length \* width \* heigh;

}

};

Rectangle.cpp

#include <iostream>

#include "Rectangle.h"

using namespace std;

int main()

{

double length, width, heigh;

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

{

cout << "The length is: " ;

cin >> length;

cout << "The width is: " ;

cin >> width;

cout << "The heigh is: " ;

cin >> heigh;

Rectangle rectangle1(length, width, heigh);

cout << "The volume of length " << rectangle1.length << " width " << rectangle1.width << " heigh " << rectangle1.heigh << " is " << rectangle1.getVolume() << endl;

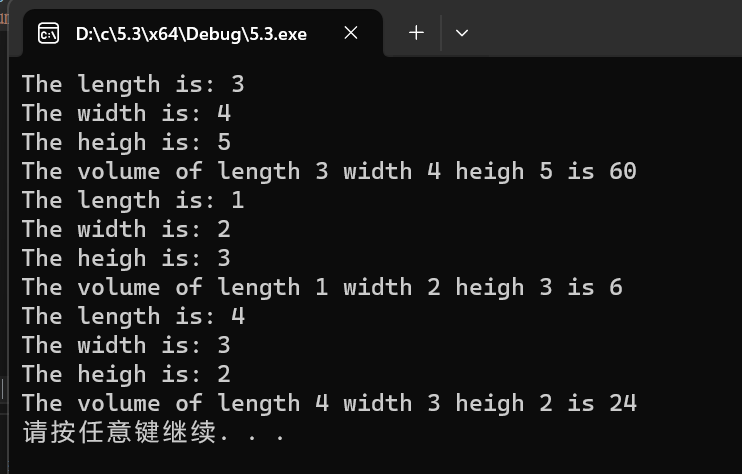
}

system("pause");

return 0;

}

结果截屏



**5.4**

实验代码

#include<iostream>

#include<math.h>

#define PI 3.14

using namespace std ;

class Shape

{

public:

double x;

double y;

virtual void getArea()

{

cout << x \* y << endl;

}

};

class Circle : public Shape{

public:

double radius;

Circle(double newradius)

{

radius = newradius;

}

virtual void getArea()

{

cout << "The area of Circle is " << radius \* radius \* PI << endl;

}

};

class Rectangle : public Shape{

public:

double length;

double width;

Rectangle(double newlength, double newwidth)

{

length = newlength;

width = newwidth;

}

virtual void getArea()

{

cout << "The area of Rectangle is" << length \* width << endl;

}

};

class Square : public Shape{

public:

double breadth;

Square(double newbreadth)

{

breadth = newbreadth;

}

virtual void getArea()

{

cout << "The area of Square is " << breadth \* breadth << endl;

}

};

class Trapezoid : public Shape{

public:

double up;

double down;

double h;

Trapezoid(double newup, double newdown, double newh)

{

up = newup;

down = newdown;

h = newh;

}

virtual void getArea()

{

cout << "The area of Trapezoid is" << (up + down) \* h / 2 << endl;

}

};

class Triangle : public Shape{

public:

double a;

double b;

double c;

Triangle(double newa, double newb, double newc)

{

a = newa;

b = newb;

c = newc;

}

virtual void getArea()

{

cout << "The area of Triangle is" << (sqrt((a + b + c) \* (a + b - c) \* (a + c - b) \* (b + c - a))) / 4 <<

endl;

}

};

int main()

{

Circle circle(12);

Square square(4);

Rectangle rectangle(1, 4);

Trapezoid trapezoid(2.0, 4.5, 3);

Triangle triangle(3, 5, 6);

circle.getArea();

square.getArea();

rectangle.getArea();

trapezoid.getArea();

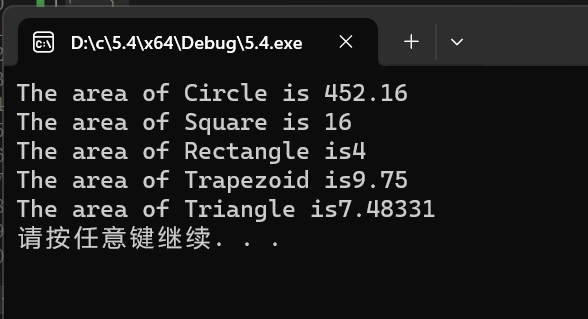
triangle.getArea();

system("pause");

return 0;

}

结果截屏



**5.5**

实验代码

#include<iostream>

#include<iomanip>

#include<vector>

#include<unordered\_map>

using namespace std;

class Solution {

public:

vector<int> intersect(vector<int>& nums1, vector<int>& nums2) {

unordered\_map<int, int> buf;

int n1 = nums1.size();

int n2 = nums2.size();

vector<int> res;

for (int i = 0; i < n1; i++)

{

buf[nums1[i]]++;

}

for (int j = 0; j < n2; j++)

{

if (buf[nums2[j]] > 0)

{

res.push\_back(nums2[j]);

buf[nums2[j]] = 0;

}

}

return res;

}

};

int main()

{

Solution s;

vector<int> num1 = { 1,2,2,1 };

vector<int> num2 = { 2,2 };

vector<int> res = s.intersect(num1, num2);

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

cout << res[i] << endl;

system("pause");

return 0;

}

结果截屏  


四、遇到的问题与解决办法

实验中，我又到很多不会的的地方，许多知识掌握的不熟练，尤其是类这一部分，掌握的尤其有问题；在做 2.9 实验时，我把日期题目理解错了，在同学们的帮助下我才找到自己的错误。在做5.1 实验时，==这个逻辑错误我找了很久，当时始终无法输出结果，经过调试，发现问题出在了那里，逻辑出了问题，陷入了死循环，那道实验题花了我真的很长时间。在做实验 5.3 时，我也犯了很严重的错误，许多问题，得不到我想要的结果，最后发现问题出现在函数调用上。还有类，const 问题，同学的帮助下解决了。  
五、体会  
做了这么多 C++的实验，感觉自己的编程的确进步了很多，最初时我编程还会犯少；(这类错误，现在，基本不会犯这种低级错误了，真的进步了很多。从最初的 cout cin,函数到数组指针类向量，感觉自己的能力在一步步地提升。在这麽多的实验中我体会到了规范和模块化的重要，最初，我还不以为然，知道吃了很多次亏后，我才发现，做题要现有宏观建构，构建相应的模块函数，在借此一步步实现，这样才不至于太过慌乱。我还理解了调试的重要，通过调试，我可以快速地发现我的错误，一步一步地去寻找，找到自己的逻辑错误，太厉害了。还有我感到通过自己的实验，我对代码理解能力上升了很多，也理解了编程是个技巧活，一定要有方法，在编程过程，我也将书上的代码熟悉了，能力真的得到了很大的提升，希望以后能更好。