**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。

1、代码：

#include<iostream>

using namespace std;

class Time //声明类

{

public: //成员函数为公有的

void set\_time() //类体内定义成员函数

{

cout << "请依次输入小时、分钟、秒" << endl;

cin >> hour;

cin >> minute;

cin >> sec;

}

void get\_time() //类体内定义成员函数

{

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

}

private: //数据成员为私有的

int hour;

int minute;

int sec;

};

Time t; //定义类对象

int main()

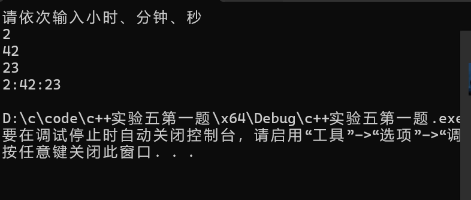
{

t.set\_time(); //调用对象的成员函数

t.get\_time(); //调用对象的成员函数

return 0;

}

结果：

}

问题：

1、设置时间和获取时间的成员函数应该被设为公用，用来和private区域的数据交互

2、hour、minute 和 sec 这些表示时间的变量应该被设为私有，防止外部代码直接修改这些值，保证稳定性。

3、简单或者基本的可以放在类内比较简洁

4、复杂的可以放在类外更好调试修改代码

第二题：

#ifndef STUDENT\_H

#define STUDENT\_H

class Student

{

public: //公用成员函数原型声明

Student(); // 默认构造函数

void set\_value(int n, const char\* nm, char s); // 设置数据成员的函数

void display() const; // 显示学生信息

private:

int num;

char name[20];

char sex;

};

#endif

#include <iostream>

#include <cstring>

#include "student.h" //不要漏写此行，否则编译通不过

using namespace std;

Student::Student() : num(0), sex(' ')

{

name[0] = '\0'; // 初始化 name 为一个空字符串

}

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

{

num = n;

// 使用 strncpy 以确保不会超出 name 数组大小

strncpy\_s(name, nm, sizeof(name) - 1);

name[sizeof(name) - 1] = '\0'; // 确保 name 数组终止符是'\0'

sex = s;

}

void Student::display() const //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()

{

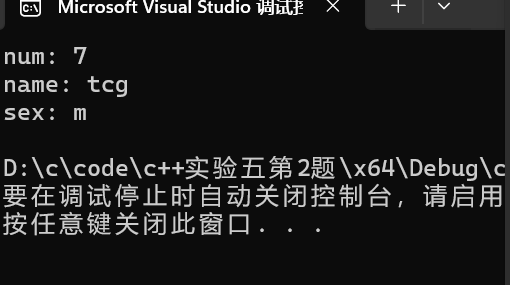
Student stud;

stud.set\_value(007, "tcg", 'm'); // 使用 set\_value 为学生对象赋值

stud.display();

return 0;

}

结果：

第三题：

代码：

#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();

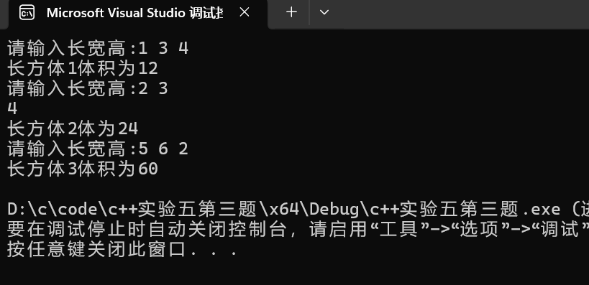
box3.get\_value();

cout<<"volume of box3 is ";

box3.display();

return 0;

}

结果：

第四题：

代码：

#include <iostream>

using namespace std;

class Student

{

public:

Student(string n, double s) : num(n), score(s) {}

string num;

double score;

};

int main()

{

void max(Student\*);

Student s[5] =

{

Student("1",93),

Student("2",100),

Student("3",67),

Student("4",82),

Student("5",95)

};

max(s);

return 0;

}

void max(Student\* p)

{

Student\* t;

Student\* c;

double max;

for (t = p, c = t, max = t->score; t < (p + 5); ++t)

{

if (t->score > max)

{

max = t->score;

c = t;

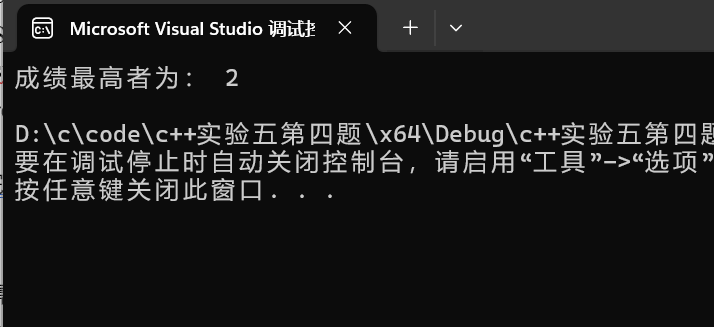
}

}

cout << "成绩最高者为： " << c->num << endl;

}

结果：



第五题：

代码：

#include <iostream>

using namespace std;

class Point {

private:

int x, y;

public:

Point() : x(60), y(80) {}

void setPoint(int i, int j)

{

x = 60 + i;

y = 80 + j;

}

void display()

{

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

}

};

int main() {

Point p;

cout << "初始坐标: ";

p.display();

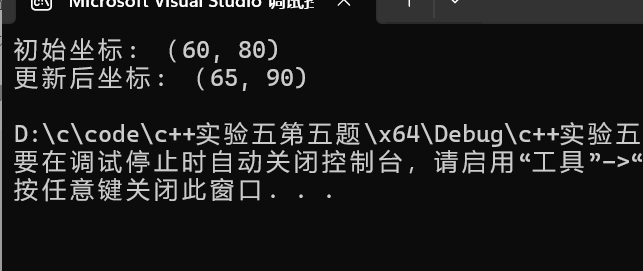
p.setPoint(5, 10);

cout << "更新后坐标: ";

p.display();

return 0;

}

结果

**问题及解决方法：**

1、第一天最初把hour、minute、sec的输入和输出全部分别做成了共六个成员函数，但后来想到事实上只需要输入和输出两个函数分别包括三个数据就可以。

**2、**第二题**中**strncpy显示什么版本报错。改成strncpy\_s就好了

3、第二题头文件的预处理指令搞错了，导致几个文件没关联起来和报错，改了就好了。

4、第五题对题意理解不够明确，有些地方没看懂，分开查了一些具体的步骤后好一点点。

**体会：**

1、实验中将数据成员设置为私有，并通过公有成员函数进行访问和修改，这增强了数据的安全性和隐藏性，防止了外部直接访问和修改内部数据，体现了数据封装的必要性。所以数据是不是最好要放在private下。这样可以自由控制数据的读写权限。

2、要注意除了顺着题目的思路走，也要试着总结规律，可能可以发现一些简化代码的地方。

3、第二题中多个文件让我意识到代码模块化的重要性，它有助于代码的组织和重用，也使得代码的维护和扩展变得更加容易。

4、有一些报错并不一定是语法问题，可能是有一些特殊的小点或者版本问题，可以试着去查一查得到结果。