**实验五 类与对象**

**一、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。

**二、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。

**三、算法分析，程序结果**

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?

**结果：**

class Time // 定义Time类

{

private: // 数据成员为公用的

int hour;

int minute;

int sec;

public:

void setTime(int h, int m, int s) {

hour = h;

minute = m;

sec=s;

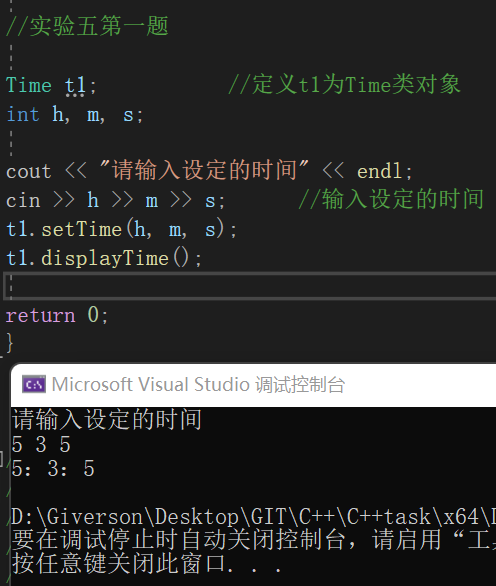
}

void displayTime() {

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

}

};



**分析：**

在面向对象编程中，对成员和函数进行公有（public）和私有（private）的适当指定是良好的设计实践。以下是一些建议：

**成员的公有和私有指定：**

**1.公有成员：**

允许外部代码访问的接口应该是公有的。这通常包括类的外部需要使用的方法、属性和一些操作函数。

提供一些获取成员变量的方法（getter）通常是公有的，以便外部代码可以读取对象的状态。

**2.私有成员：**

内部实现细节和状态信息通常是私有的，以隐藏对象的内部实现，从而提高封装性。成员变量应该被声明为私有，通过公有的方法进行访问和修改。

**函数的定义位置：**

**1.在类中定义的函数：**

类的接口（公有成员函数）应该包含高层次的操作，它们定义类的行为，提供对类的功能的访问。

将需要访问私有成员的方法放在类的内部，以确保对内部状态的修改是通过受控制的方式进行的。

构造函数和析构函数通常应在类中定义，以初始化和清理对象的状态。

**2.在类外定义的函数：**

如果一个函数的实现与类的具体实现无关，可以考虑将其定义在类外。这通常包括一些与类直接交互的全局函数。

非成员函数（不属于任何类的函数）以及一些独立于类的操作，可能更适合在类外定义。

**一些额外的建议：**

**友元函数：** 如果某个函数需要访问类的私有成员，但不是成员函数，可以将其声明为友元函数。友元函数在类中声明，在类外定义。

**接口和实现分离：** 尽量将类的接口和实现分离，以提高代码的可维护性。类的用户只需关心接口，而不必了解实现的细节。

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。

（1）Student.h

class Student //类声明

{

public: //公用成员函数原型声明

void display();

void set\_value(int num,const char \* name,char sex);

private:

int num;

char name[20];

char sex;

};

（2）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;

}

void Student::set\_value(int num, const char \* name, char sex) {

this->num = num;

strncpy\_s(this->name, name, 19);

this->name[19] = '\0';

this->sex = sex;

};

（3）main.cpp

#include <iostream> //将类声明头文件包含进来

#include “student.h”

int main()

{

Student stud; //定义对象

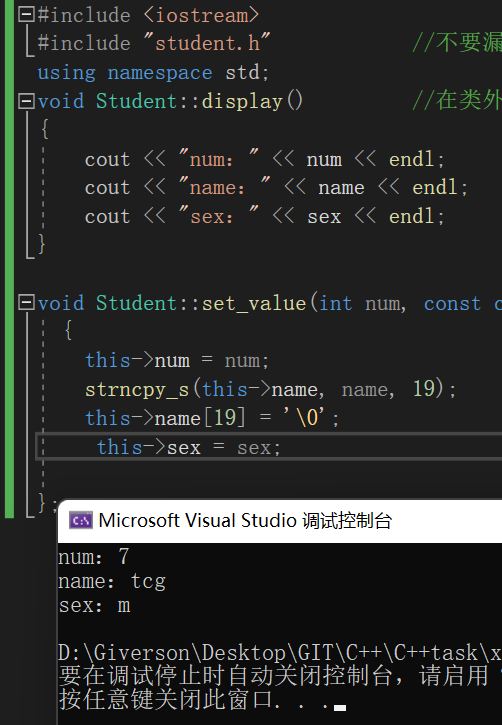
stud.set\_value(007, "tcg", 'm');

stud.display(); //执行stud对象的display函数

return 0;

}

**结果：**



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。

（1）Cuboid 类

class Cuboid {

public:

void set\_value();

void display();

private:

int length;

int wideth;

int height;

};

（2）Cuboid类函数的实现

void Cuboid::display() {

cout << "体积：" << length\*wideth\*height<< endl;

}

void Cuboid::set\_value() {

cout << "请输入长方体的长宽高：";

cin>>

this->height >>

this->length >>

this->wideth ;

}

（3）main函数

Int main(){

Cuboid c1;

Cuboid c2;

Cuboid c3;

c1.set\_value();

c2.set\_value();

c3.set\_value();

c1.display();

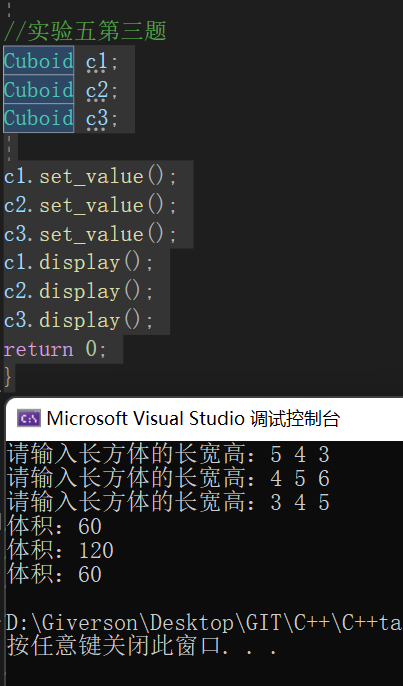
c2.display();

c3.display();

return 0;

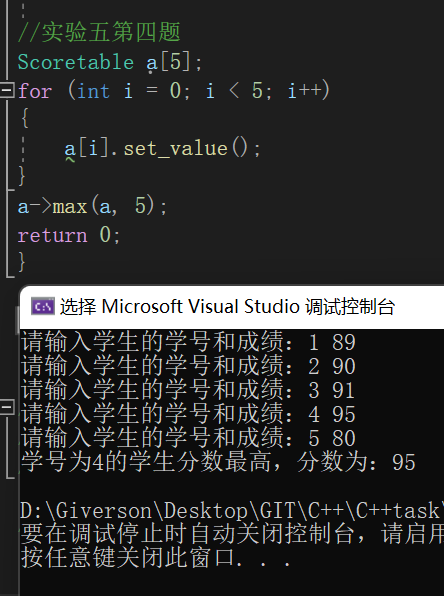
}

**结果：**



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。

**结果：**



（1）Scoretable类

class Scoretable {

public:

void max(Scoretable\* p,int n);

void set\_value();

private:

int sno;

int score;

};

（2）Scoretable函数的实现

void Scoretable::max(Scoretable\* p,int n) {

Scoretable\* max = p; //创建对象指针

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

if (p[i].score > max->score) {

max = &p[i];

}

}

cout << "学号为" <<max->sno << "的学生分数最高，分数为：" <<max->score << endl;

}

void Scoretable::set\_value() {

cout << "请输入学生的学号和成绩：";

cin >>

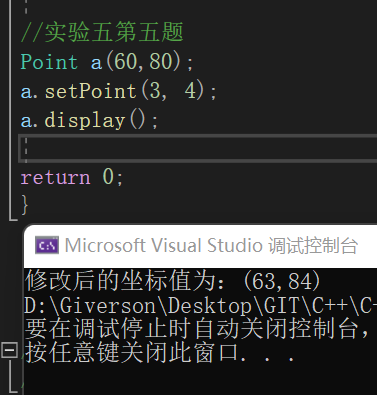
this->sno >>

this->score ;

}

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。

**结果：**



（1）Point类

class Point

{

public:

Point(int x,int y);

void setPoint(int i, int j);

void display();

private:

int x;

int y;

};

（2）Point函数的实现

Point::Point(int x,int y)

{

this->x = x;

this->y = y;

}

void Point::setPoint(int i, int j) {

this->x += i;

this->y += j;

}

void Point::display() {

cout << "修改后的坐标值为：" << "(" << this->x << "," << this->y << ")";

}

**四、遇到的问题与解决方法**

问题1、第一题，忘记了类的公有成员与私有成员的作用域范围。

解决方法：网上搜索资料，弄清两者之间的差别后解决；

问题2、第二题set\_value函数不知道如何给char[20] name赋值

解决方法：利用strncpy\_s函数，直接把字符串复制过来；

问题3、利用指针去输出数组里面对象的成员最大值有点忘了

解决方法：网上查询相关示例后解决。

**五、体会**

**（一）感想**

1、类的私有与公有成员，函数之间的作用域一定要记得，否则可能造成“函数失效”的假象。

2、有些基本的函数还是要记住，比如字符串的复制等等；

3、有一些算法，还是要重新熟悉一下，免得要用的时候又忘记了怎么写；

**（二）体会**

**1、理解类和对象的概念：**

实验过程中，我深入理解了什么是类和对象。类是一种数据结构的抽象，而对象是类的实例。通过定义类，我们可以创建多个对象，每个对象都有自己的属性和行为。

**2、声明类的方法：**

我学到了如何声明一个类，包括如何定义类的成员变量和成员函数。这是创建具有一致结构和行为的对象的关键步骤。

**3、定义对象的方法：**

通过实验，我熟悉了如何定义对象，即如何根据已声明的类创建实际的实例。这使我能够在程序中使用这些对象，并调用它们的方法。

**4、基于对象的程序设计：**

利用类和对象，我编写了基于对象的程序。这种方式的编程更加模块化和可维护，因为每个类都负责自己的功能，对象之间的交互更清晰。

**5、检查和调试基于对象的程序：**

在实验中，我学到了如何检查和调试基于对象的程序。这包括了理解对象的状态、检查对象的方法调用以及解决可能的错误。

总体而言，通过这个实验，我对面向对象编程的基本概念有了更深刻的理解，掌握了声明类、定义对象、编写基于对象的程序以及调试这些程序的基本技能。这为我以后更复杂的面向对象编程任务奠定了基础。