**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。

**一．实验报告**

1**.#include<iostream>**

**using namespace std;**

**class Time**

**{**

**private:**

**int hour;**

**int minute;**

**int sec;**

**public:**

**void input()**

**{**

**cin >> hour;**

**cin >> minute;**

**cin >>sec;**

**}**

**void output()**

**{**

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

**}**

**};**

**int main()**

**{**

**Time t1;**

**t1.input();**

**t1.output();**

**return 0;**

**}**

**那些需要被类外部的代码直接访问和操作的成员应该指定为公用的。**

**数据成员通常最好指定为私有的，像  Time  类中的  hour 、 minute 、 sec ，将它们设为私有可以对其进行有效的封装，防止外部随意修改，保证数据的完整性和安全性。**

**那些和类的某个具体操作紧密相关，功能相对简单且代码量不大的函数最好放在类中定义，比如上面改写后的  input  和output函数。**

**当函数的实现相对复杂，代码量比较大，或者函数涉及到多个类之间的交互等更复杂的逻辑时，通常放在类外定义比较好。**

****

**2.**

**//**//student.h

class Student

{

public:

void display();

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

private:

int num;

char name[20];

char sex;

};

//student.cpp

**#include <iostream>**

**#include "student.h"**

**#include<cstring>**

**using namespace std;**

**void Student::display()**

**{**

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

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

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

**}**

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

**{**

**num = n;**

**strcpy\_s(name, nm);**

**sex = s;**

**}**

//main.cpp

#include <iostream>

#include "student.h"

int main()

{

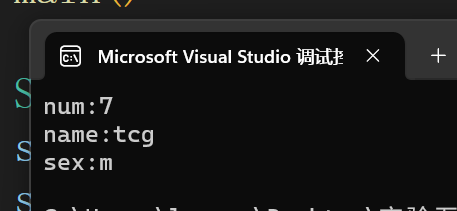
Student stud;

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

stud.display();

return 0;

}



3.#include<iostream>

using namespace std;

class cft

{

private:

int length;

int width;

int height;

public:

void input()

{

cin >> length >> width >> height;

}

void output()

{

cout << length \* width \* height << endl;

}

};

int main()

{

cft cft1;

cft cft2;

cft cft3;

cft1.input();

cft2.input();

cft3.input();

cft1.output();

cft2.output();

cft3.output();

return 0;

}



4.#include<iostream>

using namespace std;

class Student

{

private:

int num;

int score;

public:

Student(int no,int cj):num(no),score(cj){}

int getnum() const

{

return num;

}

int getscore() const

{

return score;

}

void max(Student\*students[],int size)

{

int maxt = 0;

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

{

if (students[i]->getscore() > students[maxt]->getscore())

{

maxt = i;

}

}

cout << students[maxt]->getnum() << endl;

}

};

int main()

{

Student\* students[5];

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

{

int num, score;

cin >> num >> score;

students[i] = new Student(num, score);

}

students[0]->max(students, 5);

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

{

delete students[i];

}

return 0;

}



5.**#include<iostream>**

**using namespace std;**

**class Point**

**{**

**private:**

**int x;**

**int y;**

**public:**

**void point(int x=0,int y=0)**

**{**

**this->x = x, this->y = y;**

**}**

**void setPoint(int i, int j)**

**{**

**x += i;**

**y += j;**

**}**

**void display()**

**{**

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

**}**

**};**

**int main()**

**{**

**int i, j;**

**cin >> i >> j;**

**Point point1;**

**point1.point(60, 80);**

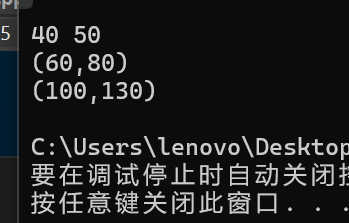
**point1.display();**

**point1.setPoint(i,j);**

**point1.display();**

**return 0;**

**}**

****

**二．遇到的问题与解决方法**

**问题：对类的声明，类的对象和成员函数的运用不熟练，对对象数组的运用不熟悉。**

**方法：上网查找，翻书**

1. **体会**
2. **对类的运用，对象数组的运用有了深刻的了解，能够基本运用并解决问题**