**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?

显示对象信息的函数、设置对象信息的函数

通常包括数据成员和实现细节相关的函数

把与类相关的函数都放在类中定义

类无关的函数，例如一些辅助函数、全局函数等，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。

#include<iostream>

using namespace std;

class Time // 定义Time类

{

private: // 数据成员为公用的

int hour;

int minute;

int sec;

public:

void setime() {

cin >> hour >> minute >> sec;

}

void display() {

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

}

};

int main()

{

Time t1;

t1.setime();

t1.display();

return 0;

}

#pragma once

//student.h

class Student //类声明

{

public: //公用成员函数原型声明

void set\_value(int n, const char c[20], char s);

void display();

private:

int num;

char name[20];

char sex;

}

#include <iostream>

#include"student.h"

using namespace std;//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;

}

void Student::set\_value(int n, const char c[20], char s) //在类外定义set\_value类函数

{

num = n;

//strcpy(name, na);

for (int i = 0; i <= 20; i++)

name[i] = c[i];

sex = s;

}

//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

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

stud.display(); //执行stud对象的display函数

return 0;

}

#include<iostream>

using namespace std;

class changfangti {

private:

double length, width, height;

public:

void volume(double l,double w,double h) {

length = l;

width = w;

height = h;

}

double count() {

return length \* width \* height;

}

};

int main()

{

changfangti box1, box2, box3;

double l, w, h;

cout << "请输入第一个长方体的三边： ";

cin >> l >> w >> h;

box1.volume(l, w, h);

cout << "请输入第二个长方体的三边： ";

cin >> l >> w >> h;

box2.volume(l, w, h);

cout << "请输入第三个长方体的三边： ";

cin >> l >> w >> h;

box3.volume(l, w, h);

double a1 = box1.count();

double a2 = box2.count();

double a3 = box3.count();

cout << "第一，二，三个长方体的体积为：" << a1 << " "

<< a2 << " " << a3 << endl;

}

#include <iostream>

using namespace std;

class student {

public:

int score;

int id;

};

void max(student\* students, int size) {

int maxscore = 0;

int maxid = 0;

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

if (students[i].score > maxscore) {

maxscore = students[i].score;

maxid = students[i].id;

}

}

cout<< "最高分数的学生学号为：" << maxid << endl;

}

int main()

{

student students[5];

students[0].id = 1;

students[0].score = 90;

students[1].id = 2;

students[1].score = 85;

students[2].id = 3;

students[2].score = 92;

students[3].id = 4;

students[3].score = 88;

students[4].id = 5;

students[4].score = 91;

max(students, 5);

return 0;

}

#include <iostream>

using namespace std;

class point {

private:

int x, y;

public:

point(int x = 60, int y = 80) {

this->x = x;

this->y = y;

}

void point1(int i, int j)

{

x = 60 + i;

y = 80 + j;

}void display()

{

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

}

};

int main() {

point p;

p.display();

p.point1(10, 20);

p.display();

return 0;

}