**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：

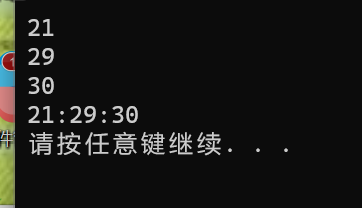
(1)将数据成员改为私有的；

(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?

****可以在类外被任意访问的成员应该定义为公有，不希望在类外被随意使用的成员应该定义为私有，简单的函数最好在类中定义，代码比较长的函数最好在外面定义****。



#include<iostream>

using namespace std;

class Time // 定义Time类

{

private: // 数据成员为私有的

int hour;

int minute;

int second;

public:

void input() // 输入

{

cin >> this->hour;

cin >> this->minute;

cin >> this->second;

}

void output() // 输出

{

cout << this->hour << ":" << this->minute << ":" << this->second << endl;

}

};

int main()

{

Time t1; //定义t1为Time类对象

t1.input(); // 输入设定时间

t1.output(); // 输出时间

system("pause");

return 0;

}

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

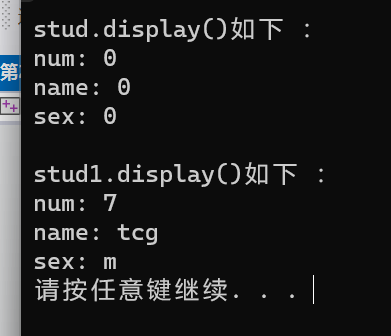
Student stud1(007,”tcg”,’m’);

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。



Student.h:

#pragma once

#include<iostream>

using namespace std;

class Student

{

public:

Student(){}

Student(int num1, string name1, char sex1);

void display();

void set\_value();

private:

int num;

string name;

char sex;

};

Student.cpp:

#include"student.h"

Student::Student(int num1, string name1, char sex1)

{

this->num = num1;

this->name = name1;

this->sex = sex1;

}

void Student::display()

{

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

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

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

}

void Student::set\_value()

{

this->num = 0;

this->name = "0";

this->sex = '0';

}

Main.cpp:

#include<iostream>

using namespace std;

#include"student.h"

// 增加一个对数据成员赋初值的成员函数set\_value

int main() {

Student stud; //定义对象

stud.set\_value();

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

cout << "stud.display()如下 ：" << endl;

stud.display(); //执行stud对象的display函数

cout << endl;

cout << "stud1.display()如下 ：" << endl;

stud1.display();

system("pause");

return 0;

}

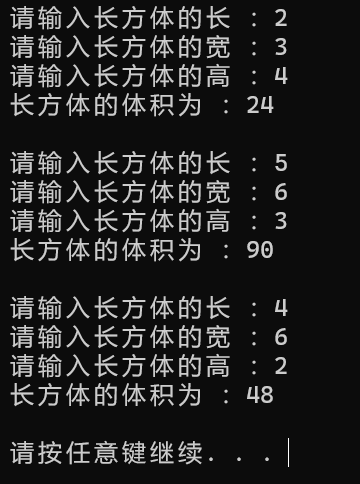
3、需要求3个长方柱的体积，请编一个基于对象的程序。数据成员包括length(长)、width(宽)、height(高)。要求用成员函数实现以下功能：

(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。



#include<iostream>

using namespace std;

class Cuboid

{

public:

int length;

int width;

int height;

void input()

{

cout << "请输入长方体的长 ：";

cin >> this->length;

cout << "请输入长方体的宽 ：";

cin >> this->width;

cout << "请输入长方体的高 ：";

cin >> this->height;

}

int calculate()

{

return this->length \* this->width \* this->height;

}

void output()

{

cout << "长方体的体积为 ：" << this->calculate() << endl;

}

};

int main() {

Cuboid c1;

c1.input();

c1.output();

cout << endl;

Cuboid c2;

c2.input();

c2.output();

cout << endl;

Cuboid c3;

c3.input();

c3.output();

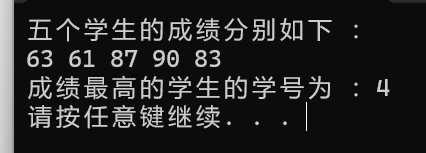
cout << endl;

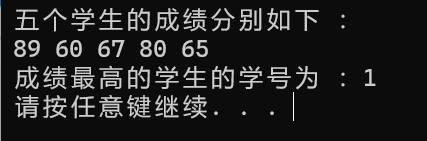
system("pause");

return 0;

}

1. 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。





#include<iostream>

using namespace std;

class student

{

public:

int m\_id; // 学号

int m\_score; // 分数

student(int id)

{

this->m\_id = id;

this->m\_score = rand() % 41 + 60;

}

};

int my\_max(student\* p, student arr[5])

{

int max = p->m\_id;

int maxscore = p->m\_score;

int count = 0;

for (student\* p = arr; count < 5; p++)

{

if (maxscore < (p + 1)->m\_score)

{

maxscore = (p + 1)->m\_score;

max = (p + 1)->m\_id;

}

count++;

}

return max;

}

int main()

{

srand((unsigned int)time(NULL));

student s1(1);

student s2(2);

student s3(3);

student s4(4);

student s5(5);

cout << "五个学生的成绩分别如下 ：" << endl;

cout << s1.m\_score << " " << s2.m\_score << " " << s3.m\_score << " "

<< s4.m\_score << " " << s5.m\_score << endl;

student arr[5] = { s1,s2,s3,s4,s5 };

student\* p = arr;

cout << "成绩最高的学生的学号为 ：";

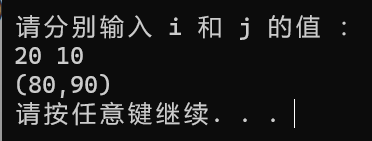
cout << my\_max(p, arr) << endl;

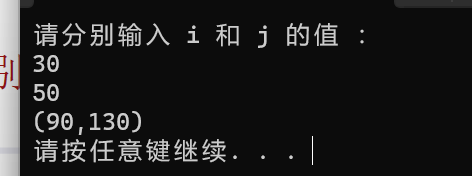
system("pause");

return 0;

}

1. 设有一描述坐标点的类Point，其私有变量x和y代表一个s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 point

{

private:

int x;

int y;

public:

point()

{

this->x = 60;

this->y = 80;

}

void setPoint(int i, int j)

{

x += i;

y += j;

}

void display()

{

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

}

};

int main()

{

point p;

int i, j;

cout << "请分别输入 i 和 j 的值 ：" << endl;

cin >> i >> j;

p.setPoint(i, j);

p.display();

system("pause");

return 0;

}