**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.算法分析，程序结果**

1. #include<iostream>

using namespace std;

class Time

{public:

void set(int hour,int minute,int sec)

{

this->hour = hour;

this->minute = minute;

this->sec = sec;

}

void show()

{

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

}

private:

int hour;

int minute;

int sec;

};

void test01()

{

Time t1;

t1.set(11,22,33);

t1.show();

}

int main()

{

test01();

/\*Time t1;

cin >> t1.hour;

cin >> t1.minute;

cin >> t1.sec;

cout << t1.hour << ":" << t1.minute << ":" << t1.sec << endl;\*/

return 0;

}

2. #pragma once

#include<iostream>

using namespace std;

class Student

{

public:

void set\_value();

void display();

private:

int num;

char name[20];

char sex;

};

#include"student.h"

void Student::display()

{

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

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

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

}

void Student::set\_value()

{

cin>> num;

cin>> name;

cin>>sex;

}

#include "student.h"

int main()

{

Student stud;

stud.set\_value();

stud.display();

return 0;

}

3. #include<iostream>

using namespace std;

class Cube

{

public:

void set()

{

cout << "长方体的长，宽。高分别为";

cin >> length;

cin >> width;

cin >> height;

v = length \* width \* height;

}

void show()

{

cout <<"长方体体积="<< v;

}

private:

int length;

int width;

int height;

int v;

};

void test01()

{

Cube c1;

c1.set();

c1.show();

}

int main()

{

for(int i=0;i<3;i++)

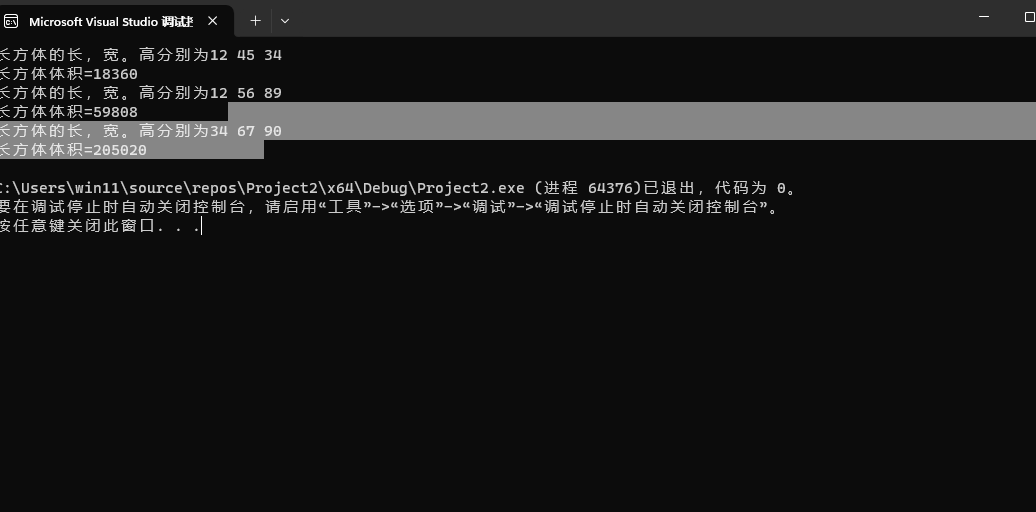
{

test01();

}

return 0;

}



4. #include<iostream>

using namespace std;

class Student

{

public:

Student(int a,int b):num(a),grade(b)

{}

int num;

int grade;

};

void Max(Student \*s)

{

int max = s[0].grade;

for (int i = 1;i < 5;i++)

{

if(s[i].grade>max)

{

max = s[i].grade;

}

}

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

{

if (s[i].grade == max)

{

cout << s[i].num << endl;

}

}

}

void test01()

{

Student s[5] = { {1,99},{2,95},{3,93},{4,89},{5,91} };

/\*s[0].num = 1;

s[0].grade = 99;

s[1].num = 2;

s[1].grade = 96;

s[2].num = 3;

s[2].grade = 89;

s[3].num = 4;

s[3].grade = 93;

s[4].num = 5;

s[4].grade = 95;\*/

Max(s);

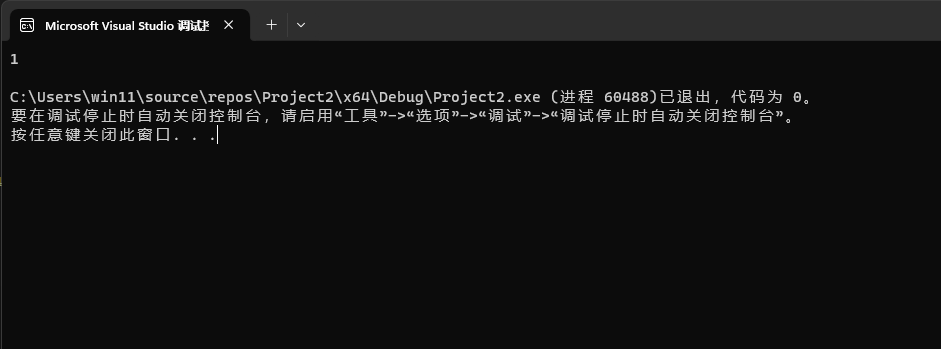
}

int main()

{

test01();

}



5. #include<iostream>

using namespace std;

class Point

{

public:

Point()

{

x = 60;

y = 80;

}

void setpoint(int i, int j)

{

x += i;

y += j;

}

void display()

{

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

}

private:

int x;

int y;

};

int main()

{

Point p;

p.display();

p.setpoint(20, 50);

p.display();

}

