**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。

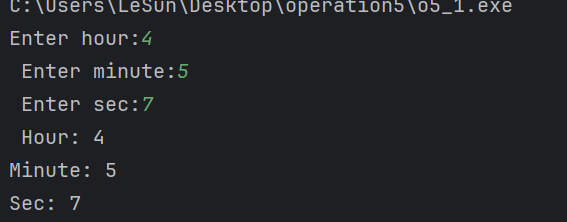
请编程序，上机调试并运行。

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。

代码展示：

1. 代码：
2. #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 input(){  
    cout<<"Enter hour: ";  
    cin>>hour;  
    cout<<"Enter minute: ";  
    cin>>minute;  
    cout<<"Enter sec: ";  
    cin>>sec;  
    }  
    void show(){  
    cout<<"Hour: "<<hour<<endl;  
    cout<<"Minute: "<<minute<<endl;  
    cout<<"Sec: "<<sec<<endl;  
    }  
   };  
   int main()  
   {  
    Time t;  
    t.input();  
    t.show();  
    return 0;  
   }

运行结果：

2，代码：

Main.h:

*///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;  
 stud1.set\_value(007, "tcg",'m'); // 使用新增的函数给stud1对象赋值  
 stud.display(); //执行stud对象的display函数  
 stud1.display();  
 return 0;  
}

Student.cpp:

//student.cpp 在此文件中进行函数的定义  
#include <iostream>  
#include<string>  
#include<cstring>  
#include "student.h" //不要漏写此行，否则编译通不过  
using namespace std;  
// 在类外定义display类函数  
void Student::display()  
{  
 std::cout << "num：" << num << std::endl;  
 std::cout << "name：" << name << std::endl;  
 std::cout << "sex：" << sex << std::endl;  
}  
  
// 在类外定义set\_value类函数，实现给数据成员赋值  
void Student::set\_value(int n, const char\* na, char s)  
{  
 num = n;  
 strcpy(name, na);  
 sex = s;  
}

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();  
 void set\_value(int n, const char\* na, char s); // 新增的设置值的函数声明  
private:  
 int num;  
 char name[20];  
 char sex;  
};

1. 代码：

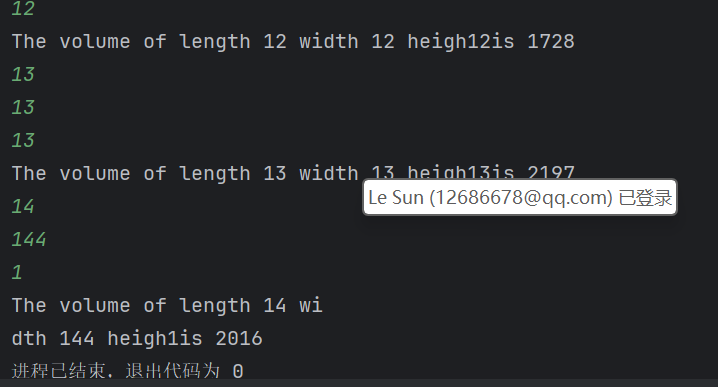
头文件：

class Rectangle  
{ public:  
 Rectangle()  
 {  
 length = 1;  
 width = 1;  
 heigh = 1;  
 }  
 Rectangle(double newLength,double newWidth,double newHeigh)  
 {  
 length = newLength;  
 width = newWidth;  
 heigh = newHeigh;  
  
 }  
  
 double length;  
 double width;  
 double heigh;  
 double getVolume()  
 {  
 return length\*width\*heigh;  
 }  
  
};

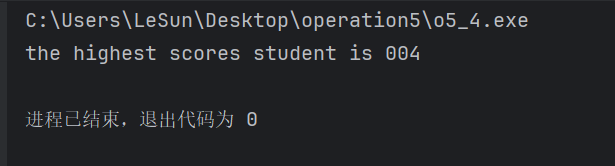
cpp文件：

#include <iostream>  
#include "Rectangle.h"  
using namespace std;  
int main()  
{  
 double length,width,heigh;  
 for(int i=0;i<3;i++)  
 {  
 cin>>length>>width>>heigh;  
 Rectangle rectangle1(length,width,heigh);  
 cout<<"The volume of length "<<rectangle1.length<<" width "<<rectangle1.width<<" heigh"<<rectangle1.heigh<<"is "<<rectangle1.getVolume();  
 }  
}

运行结果：

4代码：

#include <iostream>  
#include <string>  
using namespace std;  
  
class Student {  
public:  
 string id; // 学号  
 int score; // 成绩  
 Student(string s, int sc) : id(s), score(sc) {}  
};  
  
// max函数，使用指向对象的指针作参数来找出最高成绩者的学号  
void max(Student\* students[], int n) {  
 int maxIndex = 0;  
 for (int i = 1; i < n; ++i) {  
 if (students[i]->score > students[maxIndex]->score) {  
 maxIndex = i;  
 }  
 }  
 cout << "成绩最高的学生学号为: " << students[maxIndex]->id << endl;  
}  
  
int main() {  
 Student\* students[5];  
 students[0] = new Student("001", 85);  
 students[1] = new Student("002", 90);  
 students[2] = new Student("003", 88);  
 students[3] = new Student("004", 92);  
 students[4] = new Student("005", 80);  
  
 max(students, 5);  
  
 // 释放动态分配的内存  
 for (int i = 0; i < 5; ++i) {  
 delete students[i];  
 }  
  
 return 0;  
}



5代码：

#include <iostream>  
using namespace std;  
  
class Point {  
private:  
 int x;  
 int y;  
public:  
 // 构造函数，初始化坐标值为(60, 80)  
 Point() {  
 x = 60;  
 y = 80;  
 }  
 // 成员函数，用于修改坐标值  
 void setPoint(int i, int j) {  
 x = 60 + i;  
 y = 80 + j;  
 }  
 // 成员函数，用于显示坐标值  
 void display() {  
 cout << "(" << x << ", " << y << ")" << endl;  
 }  
};  
  
int main() {  
 Point p; // 创建Point类的对象p  
 p.display(); // 输出初始坐标值  
  
 p.setPoint(10, 20); // 修改坐标值  
 p.display(); // 输出修改后的坐标值  
  
 return 0;  
}

