**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软件工程2101

学 号： 8203210810

姓 名： 曾凡硕

**实验报告成绩：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验** | **实验一** | **实验二** | **实验三** | **实验四** | **实验五** | **总评** |
| **成绩** |  |  |  |  |  |  |

**批阅教师：**

# 实验五 类与对象

### 一、实验目的

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。

### 二、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。

### 三、算法分析、程序与结果

1、

#include<iostream>

using namespace std;

class Time {

private:

    int hour;

    int minute;

    int sec;

public:

    void setTime() {

        cin >> hour;

        cin >> minute;

        cin >> sec;

    }

    void displayTime() {

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

    }

};

int main() {

    Time t1;

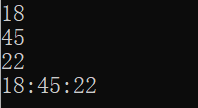
    t1.setTime();

    t1.displayTime();

    return 0;

}

运行结果：



问题：

哪些成员应指定为公共成员？

公共成员应该是那些旨在在类外部访问或修改的成员。在这种情况下，setTime和displayTime函数被指定为公共以允许外部访问。

哪些成员应指定为私有成员？

私有成员是那些不应从类外部直接访问或修改的成员。在这种情况下，hour、minute和sec数据成员被指定为私有以封装类的内部状态。

哪些函数最好放在类定义中？

直接操作类的数据成员并与类的行为密切相关的函数最好放置在类定义中。在这种情况下，setTime函数displayTime是在类内部定义的，因为它们直接操作类的私有成员Time。

哪些函数最好放置在类定义之外？

与类的内部状态不直接相关并且不需要访问其私有成员的函数最好放置在类定义之外。在这个简单的示例中，该main函数位于类外部，因为它协调程序的执行，但不直接操作类Time的内部状态。

2、

Main.cpp:

#include <iostream>

#include "student.h"

using namespace std;

int main() {

    Student stud;

    stud.set\_value(007, "tcg", 'm');

    stud.display();

    return 0;

}

Student.h:

#include<iostream>

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"

#pragma warning(disable:4996)

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;

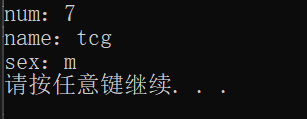
    strncpy(name, nm, 19);

    name[19] = '\0';

    sex = s;

}

运行结果：



3、

#include <iostream>

using namespace std;

class Cuboid {

private:

 int length;

 int width;

 int height;

public:

 int caculate() {

  return length \* width \* height;

 }

 void init() {

  cin >> length;

  cin >> width;

  cin >> height;

 }

};

int main() {

 Cuboid cuboid[3];

 for (int i = 0; i < 3; i++) {

  cout << "请输入第 " << i + 1 <<" 个长方体的长宽高：";

  cuboid[i].init();

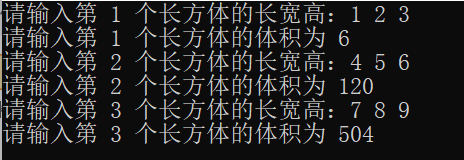
  cout << "请输入第 " << i + 1 <<" 个长方体的体积为 " << cuboid[i].caculate() << endl;

 }

 return 0;

}

运行结果：



4、

#include <iostream>

#include<string>

using namespace std;

class Student {

public:

    string studentID;

    double score;

};

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

    double maxScore = -1.0;

    int maxIndex = -1;

    for (int i = 0; i < size; ++i) {

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

            maxScore = students[i].score;

            maxIndex = i;

        }

    }

 cout << "成绩最高的学生的学号是： " << students[maxIndex].studentID << endl;

}

int main() {

    const int size = 5;

    Student students[size];

    for (int i = 0; i < size; ++i) {

        cout << "输入第 " << i + 1 << " 个学生的学号: ";

        cin >> students[i].studentID;

        cout << "输入成绩: ";

        cin >> students[i].score;

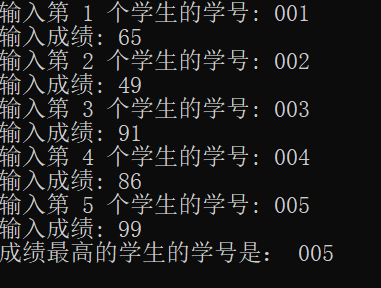
    }

    max(students, size);

    return 0;

}

运行结果：



5、

#include <iostream>

using namespace std;

class Point {

private:

    double x;

    double y;

public:

    Point() : x(60), y(80) {}

    void setPoint(double i, double j) {

        x = 60 + i;

        y = 80 + j;

    }

    void display() {

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

    }

};

int main() {

    Point myPoint;

    cout << "初始点：";

    myPoint.display();

    double dx, dy;

    cout << "输入位移：" << endl;

    cin >> dx >> dy;

    myPoint.setPoint(dx, dy);

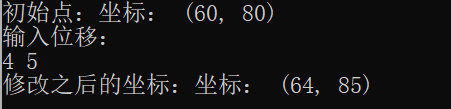
    cout << "修改之后的坐标：";

    myPoint.display();

    return 0;

}

运行结果：



### 四、遇到的问题与解决办法

第2题.字符串的复制上需要用到strcpy。另外，在student.cpp中定义函数属于类外定义，需要用到类名加作用域符合。

第4题.复习了指向对象的指针。

第5题.刚开始不知道怎么在定义对象时将x、y坐标值初始化为（60,80），复习后成功写出了代码。

### 五、体会

1、类的声明与对象的定义：类的声明涉及确定类的名称、属性（数据成员）和行为（成员函数）。实验中，通过对Time类和Student类的操作，明白了如何有效地声明类并定义其对象。尤其是在数据封装方面，我学会了如何将数据成员设为私有，以保障数据的安全性和完整性，同时使用公共成员函数来进行数据的访问和操作。

2、类成员的访问控制：通过将Time类的数据成员设置为私有并通过公共成员函数进行访问，我理解了封装的重要性。封装不仅保护了数据免受外部干扰，而且还提供了一种机制来控制数据的读写方式，这在编程实践中是非常重要的。

3、成员函数的实现与使用：在Student类中增加set\_value函数和在Cuboid类中实现体积计算的过程，使我认识到成员函数是类功能的实现者。通过成员函数，可以操作和修改对象的内部状态，这为面向对象编程提供了灵活性和功能性。

4、对象数组与指针的应用：在实验中使用对象数组来存储多个对象的信息，并通过指针来访问这些对象，让我更加熟悉了数组和指针在面向对象编程中的应用。特别是在找出最高成绩的学生时，我了解到了如何高效地通过对象数组和指针操作来实现复杂的功能。

5、构造函数与成员函数的使用：在Point类的实验中，我学会了如何使用构造函数来初始化对象，并通过成员函数改变对象状态。这强化了我对构造函数在对象创建过程中的作用的理解。