**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。

二.算法分析，程序结果

1. #include<iostream>

using namespace std;

class Time

{

private:

int hour;

int minute;

int sec;

public:

void input()

{

cout << "input hour:";

cin >> hour;

cout << endl;

cout << "input minute:";

cin >> minute;

cout << endl;

cout << "input second:";

cin >> sec;

}

void output()

{

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

}

};

int main()

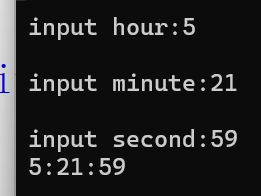
{

Time tl;

tl.input();

tl.output();

return 0;

}

①什么成员应指定为公用的-----需要外部访问的成员。

②什么成员应指定为私有的------不需要外部直接访问的。

③什么函数最好放在类中定义------那些简短且简单的函数可以放在类中定义，这样可以使代码更简单易读。

④什么函数最好在类外定义------复杂且较长的函数应该放到类之外，才可保持类定义之内比较简洁。

2.student.h:

#include <iostream>

class Student

{

public:

void display();

void set\_value(int x,const char \*arr, char z);

private:

int num;

char name[10];

char sex;

};

Student.cpp:

#include <iostream>

#include "student.h"

using namespace std;

void Student::display()

{

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

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

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

}

void Student::set\_value(int x,const char\* arr, char z)

{

num = x;

memcpy(name, arr, sizeof(name));

sex = z;

}

Main.cpp:

#include <iostream>

#include "student.h"

int main()

{

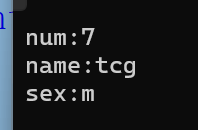
Student stud;

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

stud.display();

return 0;

}



3. #include <iostream>

using namespace std;

class cuboid

{

private:

int length, height, width;

public:

cuboid() : length(0), width(0), height(0) {};

void input()

{

cout << "请输入长方体的长:";

cin >> length;

cout << endl;

cout << "请输入长方体的高:";

cin >> height;

cout << endl;

cout << "请输入长方体的宽:";

cin >> width;

}

int cal\_volume()

{

int volume = length \* height \* width;

return volume;

}

};

int main()

{

cuboid C1,C2,C3;

cout << "请输入第一个长方体的长宽高:"<<endl;

C1.input();

int V1 = C1.cal\_volume();

cout << endl;

cout << "请输入第二个长方体的长宽高:"<<endl;

C2.input();

int V2 = C2.cal\_volume();

cout << endl;

cout << "请输入第三个长方体的长宽高:"<<endl;

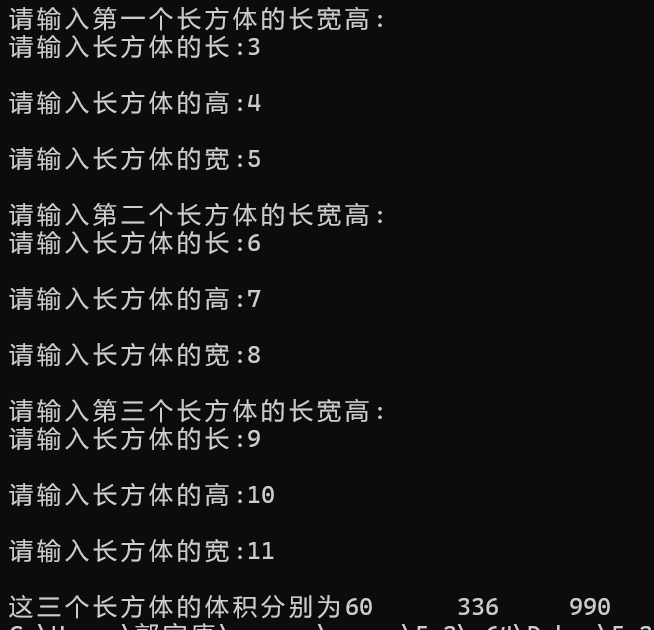
C3.input();

int V3 = C3.cal\_volume();

cout << endl;

cout << "这三个长方体的体积分别为";

cout << V1 << '\t' << V2 << '\t' << V3;

}

4. #include <iostream>

using namespace std;

class Student

{

private:

int num;

double score;

public:

void input\_num()

{

cout << "请输入学生的学号：";

cin >> num;

}

void input\_score()

{

cout << "请输入学生的成绩：";

cin >> score;

}

double return\_score()

{

return score;

}

double return\_num()

{

return num;

}

};

void max(Student\* students)

{

Student\* max\_score = &students[0];

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

{

if (students[i].return\_score() > max\_score->return\_score())

{

max\_score = &students[i];

}

}

cout << "最高成绩的学生学号是: " << (\*max\_score).return\_num()<< '\t'<<"成绩是: " << (\*max\_score).return\_score() << endl;

}

int main()

{

Student students[5];

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

{

cout << "请输入第 " << i + 1 << " 个学生的信息：" << endl;

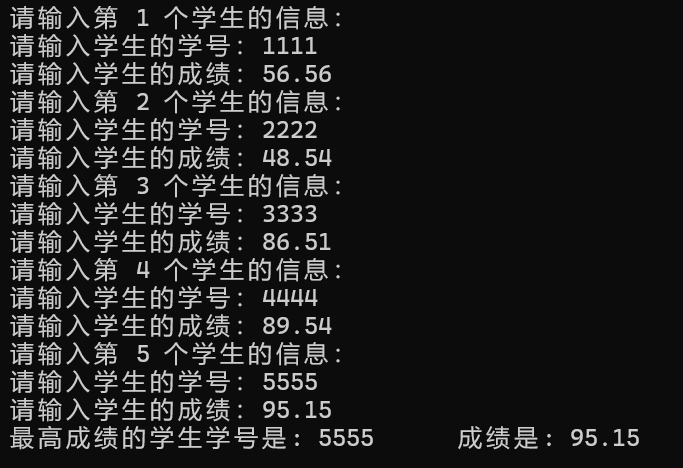
students[i].input\_num();

students[i].input\_score();

}

max(students);

}



5. #include <iostream>

using namespace std;

class point

{

private:

int x;

int y;

public:

point() :x(60), y(80) {}

void setPoint(int i, int j)

{

x = x + i;

y = y + j;

}

void display()

{

cout << "修改后的坐标值为："<<"("<<x<<","<<y<<")";

}

};

int main()

{

point p;

cout << "初始坐标值为：(60,80)" << endl;

int i = 50;

int j = 50;

p.setPoint(i, j);

p.display();

return 0;

}

