

## 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 t1;        //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.hour<<"": "<<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。

## 【实验结果】

1.

```
#include<iostream>
using namespace std;
class Time          // 定义Time类
{
private:            // 数据成员改为私有的
    int hour;
    int minute;
    int sec;
public:
    Time()          //将输入的功能改为由成员函数实现
    {
        cin >> hour >> minute >> sec;
    }
    void ShowTime() //将输出的功能改为由成员函数实现
    {
        cout << hour << ": " << minute << ": " << sec << endl;
    }
};
int main()
{
    Time t1;         //定义t1为Time类对象
    t1.ShowTime();
    return 0;
}
```



Microsoft Visual Studio 调试 × + -

12 0 0  
12: 0: 0

D:\Learning\C++\experiment\第三次实验\experiment4\5-1\x64\Debug\5-1.exe (进程 18664)已退出, 代码为 0。  
按任意键关闭此窗口。...

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？

(1) 成员函数通常指定为公用。它们将提类的接口，允许用户直接访问和操作类的私有数据；

(2) 所有的数据成员（除非它们是静态的）通常被指定为私有。这种封装意味着类的使用者不能直接访问或修改这些成员，避免对象被意外地修改；

(3) 通常将构造函数和析构函数等特殊函数在类中直接定义，小的、简单的成员函数也可以在类中定义；

(4) 对于较大的、复杂的成员函数，通常将它们的定义放在类定义之外，这样可以避免类的定义过于臃肿，以保持代码的清晰和易于维护。

2.

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 set_value(int num, const char* name, const char sex);
    void display();
private:
    int num;
    char name[20];
    char sex;
};
```

student.cpp

```
//在此文件中进行函数的定义
#include <iostream>
#include <iomanip>
#include "student.h"
using namespace std;
void Student::set_value(int num, const char* name, const char sex)
{
    this->num = num;
    int len1 = strlen(name);
    this->sex = sex;
    strcpy_s(this->name, sizeof(name) - 1, name);
    this->name[sizeof(name) - 1] = '\0';
}
void Student::display()          //在类外定义display类函数
{
    cout << "num: " << setfill('0') << setw(3) << num << endl;
    cout << "name: " << name << endl;
    cout << "sex: " << sex << endl;
}
```

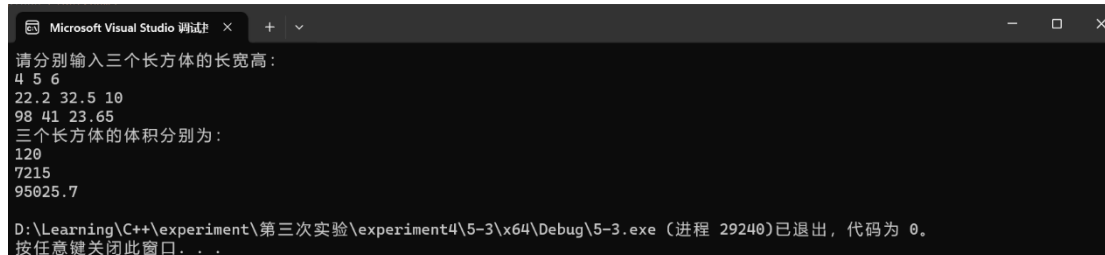
main.cpp

```
#include <iostream>
#include "student.h"
using namespace std;
int main()
{
    Student stud;
    stud.set_value(7, "tcg", 'm');          //定义对象
    stud.display();                          //执行stud对象的display函数
    return 0;
}
```

```
Microsoft Visual Studio 调试  x + v - □ x
num: 007
name: tcg
sex: m
D:\Learning\C++\experiment\第三次实验\experiment4\5-2 (1) \x64\Debug\5-2 (1) .exe (进程 11820)已退出，代码为 0。
按任意键关闭此窗口...
```

3.

```
#include <iostream>
using namespace std;
class Cuboid
{
private:
    double length, width, height;
public:
    Cuboid()
    {
        cin >> length >> width >> height;
    }
    void ShowVolume();
};
void Cuboid::ShowVolume()
{
    double volume;
    volume = length * width * height;
    cout << volume << endl;
}
int main()
{
    cout << "请分别输入三个长方体的长宽高: " << endl;
    Cuboid cuboid1;
    Cuboid cuboid2;
    Cuboid cuboid3;
    cout << "三个长方体的体积分别为: " << endl;
    cuboid1.ShowVolume();
    cuboid2.ShowVolume();
    cuboid3.ShowVolume();
    return 0;
}
```



```
Microsoft Visual Studio 调试
请分别输入三个长方体的长宽高:
4 5 6
22.2 32.5 10
98 41 23.65
三个长方体的体积分别为:
120
7215
95025.7
D:\Learning\C++\experiment\第三次实验\experiment4\5-3\x64\Debug\5-3.exe (进程 29240)已退出, 代码为 0。
按任意键关闭此窗口...
```

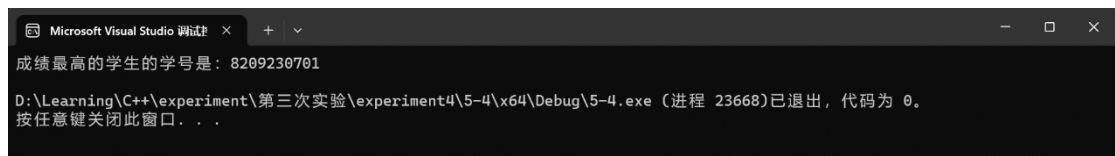
4.

```
#include <iostream>
#include <cstring>
using namespace std;
class StudentScore
{
private:
    char* id;
    char* name;
    float score;
public:
    StudentScore(const char* id = NULL, const char* name = NULL, float score = 0) //构造函数
    {
        this->id = new char[strlen(id) + 1];
        strncpy_s(this->id, strlen(id) + 1, id, strlen(id));
        this->name = new char[strlen(name) + 1];
        strncpy_s(this->name, strlen(name) + 1, name, strlen(name));
        this->score = score;
    }
    ~StudentScore() //析构函数
    {
        if (id != NULL)
        {
            delete[] id;
            id = NULL;
        }
        if (name != NULL)
        {
            delete[] name;
            name = NULL;
        }
    }

    void max(StudentScore* Stu[5]); //该函数用于输出成绩最高学生的学号
};

void StudentScore::max(StudentScore* Stu[5])
{
    float MaxScore=0;
    char* MaxScoreStudent = new char[strlen(Stu[0]->id) + 1];
    for (int i = 0; i < 5; i++)
    {
        if (MaxScore < Stu[i]->score)
        {
            MaxScore = Stu[i]->score;
            strncpy_s(MaxScoreStudent, strlen(Stu[i]->id) + 1, Stu[i]->id, strlen(Stu[i]->id));
        }
    }
    cout << "成绩最高的学生的学号是: " << MaxScoreStudent << endl;
}

int main()
{
    StudentScore stu[5] = {
        StudentScore("8209230701", "Zhang San", 92.5),
        StudentScore("8209230702", "Li Si", 90),
        StudentScore("8209230703", "Wang Wu", 77.5),
        StudentScore("8209230704", "Zhao Liu", 80),
        StudentScore("8209230705", "Liu Qi", 70),
    };
    StudentScore* Stu[5];
    for (int i = 0; i < 5; i++)
    {
        Stu[i] = &stu[i];
    }
    stu[0].max(Stu);
    return 0;
}
```



5.

```
#include <iostream>
using namespace std;
class Point
{
private:
    int x, y;
public:
    Point()
    {
        x = 60;
        y = 80;
    }

    void setPoint(int i, int j);
    void display();
};

void Point::setPoint(int i, int j)
{
    x = x + i;
    y = y + j;
}

void Point::display()
{
    cout << "修改后的坐标值为: " << "(" << x << ", " << y << ")" << endl;
}

int main()
{
    cout << "请修改坐标值: " << endl;
    int i, j;
    cout << "x值改变: ";
    cin >> i;
    cout << "y值改变: ";
    cin >> j;
    Point point;
    point.setPoint(i, j);
    point.display();
    return 0;
}
```

Microsoft Visual Studio 调试 × + -

请修改坐标值:  
x值改变: 5  
y值改变: 10  
修改后的坐标值为: (65,90)

D:\Learning\C++\experiment\第三次实验\experiment4\5-5\x64\Debug\5-5.exe (进程 8168)已退出, 代码为 0。  
按任意键关闭此窗口。 . . .