

实验四 类和对象（二）

姓 名：罗啸

学 号：2420173095

班 级：电子 173

实验地点：电气学院 412

指导老师：邝先验

4.1 实验目的

1. 理解掌握 this 指针的作用和用法;
2. 理解掌握构造函数的定义和作用;
3. 掌握构造函数的使用;
4. 理解掌握拷贝构造函数的定义和使用;
5. 理解掌握构造函数的重载;
6. 理解掌握析构函数的定义和使用。

4.2 实验内容

4.2.1 程序阅读

1. 理解下面的程序，并在 VC++6.0 下运行查看结果，回答程序后面的问题。

```
#include <iostream>
using namespace std;
class CPoint
{
public:
    void Set(int x,int y);
    void Print();
private:
    int x;
    int y;
};

void CPoint::Set(int x,int y)
{
    x = x;
    y = y;
}
void CPoint::Print()
{
    cout<<"x="<<x<<"y="<<y<<endl;
}

void main()
{
    CPoint pt;
    pt.Set(10,20);
```

```
    pt.Print();
}
```

问题一：以上程序编译能通过吗，试解释该程序？

答：

能编译通过。首先，该程序定义了一个 `CPoint` 类，类中包含两个数据成员和两个函数成员，函数成员的功能分别为为数据成员赋值和显示数据成员，但在赋值时，未赋值成功，只是对形参进行操作，故输出值为系统给 `x,y` 初始化的值。

问题二：以上程序的运行结构是否正确，如果不正确，试分析为什么，应该如何改正？

答：在赋值时，未赋值成功，只是对形参进行操作，故输出值为系统给 `x,y` 初始化的值。

将 `void CPoint::Set(int x, int y)`

```
{
    x = x;
    y = y;
}
```

改为 `void CPoint::Set(int x, int y)`

```
{
    this->x = x;
    this->y = y;
}
```

2. 理解下面的程序，并在 VC++6.0 下运行查看结果，回答程序后面的问题。

```
#include <iostream>
using namespace std;
class CPerson
{
public:
    void Print();
private:
    CPerson();
private:
    int age;
    char *name;
};
CPerson::CPerson()
{
}
void CPerson::Print()
{
```

```

        cout<<"name="<<name<<",age="<<age<<endl;
    }

```

```

void main()
{
    CPerson ps(23,"张三");
    ps.Print();
}

```

问题一：以上程序存在三处大错误，在不改变主函数内容的前提下，试改正该程序。

答：

改正后的程序：

```

#include <iostream>
#include <string>
using namespace std;
class CPerson
{
public:
    void Print();
    CPerson(int i, string a)
    {
        this->age = i;
        this->name = a;
    }

private:
    int age;
    string name;
};

void CPerson::Print()
{
    cout << "name=" << name << ",age=" << age << endl;
}

void main()
{
    CPerson ps(23, "张三");
    ps.Print();
}

```

4.2.2 程序设计

1. 设计实现一个 CPoint 类，满足以下要求：

- a. 该类包含两个整型成员变量 x (横坐标) 和 y (纵坐标), 以及一个输出函数 `Print()` 用来输出横坐标和纵坐标, 要求不可以在类的外部直接访问成员变量;
- b. 可以采用没有参数的构造函数初始化对象, 此时的成员变量采用默认值 0;
- c. 可以采用直接输入参数的方式来初始化该类的成员变量;
- d. 可以采用其它的 `CPoint` 对象来初始化该类的成员变量;
- e. 设计一个主函数来测试以上功能。

设计程序如下:

```
#include<iostream>
using namespace std;

class Point
{
private:
    int x, y;
public:
    void Print();
    Point() { x = 0; y = 0; }
    Point(int i, int j) :x(i), y(j) {};           //构造函数
};

void Point::Print()                             //输出
{
    cout << "x = " << x << "   y = " << y << endl;
}

int main()
{
    Point p1, p2(3, 4), p3;
    p3 = p2;
    p1.Print();           //验证无参数构造函数
    p2.Print();           //验证有参数构造函数
    p3.Print();           //验证类与类之间赋值
    return 0;
}
```

4.3 思考题

1. 设计一个 `CStudent` (学生) 类, 并使 `CStudent` 类具有以下特点:
 - a. 该类具有学生姓名、学号、程序设计、信号处理、数据结构三门课程的成绩;
 - b. 学生全部信息由键盘输入, 以提高程序的适应性;
 - c. 通过成员函数统计学生平均成绩, 当课程数量增加时, 成员函数无须修改仍可以求取平均成绩;
 - d. 输出学生的基本信息、各科成绩与平均成绩;

e. 学生对象的定义采用对象数组实现;

f. 统计不及格学生人数。

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class CStudent
```

```
{
```

```
private:
```

```
    int num;//学号
```

```
    string name;//姓名
```

```
    int cxsj_grad, xhcl_grad, sjjg_grad;
```

```
public:
```

```
    CStudent() { num = 1111; cxsj_grad = 0; xhcl_grad = 0; sjjg_grad = 0; name = "aaa"; }
```

```
    void xinxi_set(); //输入信息
```

```
    void Show();      //显示信息
```

```
    double average_grad();
```

```
    int jige();        //判断是否有未几个的科目，若有未及格科目返回 1，否则返回 0.
```

```
};
```

```
void CStudent::xinxi_set()
```

```
{
```

```
    cout << endl << "依次输入学号、姓名、程序设计、信号处理、数据结构三门课程的成绩,分别以回车结尾" << endl;
```

```
    cin >> num >> name >> cxsj_grad >> xhcl_grad >> sjjg_grad;
```

```
}
```

```
void CStudent::Show()      //显示信息
```

```
{
```

```
    cout << "学号: " << num << endl;
```

```
    cout << "姓名: " << name << endl;
```

```
    cout << "程序设计成绩: " << cxsj_grad << endl;
```

```
    cout << "信号处理成绩: " << xhcl_grad << endl;
```

```
    cout << "数据结构成绩: " << sjjg_grad << endl;
```

```
    cout << "平均成绩" << average_grad() << endl;
```

```
}
```

```
double CStudent::average_grad()
```

```
{
```

```
    int n = 0;
```

```
    if (this->cxsj_grad != 0) n++;
```

```
    if (this->sjjg_grad != 0) n++;
```

```
    if (this->xhcl_grad != 0) n++;
```

```
    return double(cxsj_grad + sjjg_grad + xhcl_grad) / n;
```

```

}
int CStudent::jige()
{
    if (xhcl_grad < 60 || sjjg_grad < 60 || cxsj_grad < 60)
        return 1;
    else
        return 0;
}
int main()
{
    CStudent s[3];
    for (int i = 0; i < 3; i++)
        s[i].xinxi_set();
    for (int i = 0; i < 3; i++)
        s[i].Show();
    int number=0;
    for (int i = 0; i < 3; i++)
    {
        number += s[i].jige();
    }
    cout << endl<<"不及格人数为: "<<number << endl;
    return 0;
}

```