



燕山大学  
YANSHAN UNIVERSITY

# C++面向对象程序设计 实验指导书

(3) 类与对象、构造与析构函数

燕山大学软件工程系

## 目 录

实验 3 类与对象、构造与析构函数 .....	1
1.1 时间安排 .....	1
1.2 实验目的和要求 .....	1
1.3 实验报告的撰写要求 .....	1
1.4 实验内容 .....	1



## 实验 3 类与对象、构造与析构函数

### 1.1 时间安排

本实验安排 4 个实验课时。

### 1.2 实验目的和要求

1. 掌握类的定义和使用。
2. 掌握对象的定义和对象的声明。
3. 掌握具有不同访问属性的成员的访问方式。
4. 深入体会构造函数、复制构造函数、析构函数的执行机制。

5. 使用的 VS 的断点调试功能观察程序流程，跟踪观察类的构造函数、析构函数、成员函数的执行顺序。

### 1.3 实验报告的撰写要求

将实验内容中红色字体的题目的构思过程、源码、运行结果（截图）、心得体会等内容按要求填写，详见实验报告模板。

### 1.4 实验内容

(1) 建立球体类 `sphere`。`sphere` 的构造函数要为球体的半径 `r` (`double` 型) 赋初值。用类 `sphere` 计算球体的体积 `v`。类 `sphere` 中还包含一成员函数 `vol()`，用来显示球体对象的体积。请写出完整的程序用来计算某球体的体积并输出。

球体的体积公式：
$$v = \frac{4}{3} \pi r^3$$

```
#include <iostream>
#include <cmath>
using namespace std;
class sphere
{
private:
    double r;
    double v;

public:
    sphere(double radius=0);    //构造函数;
    void vol();                 //计算并输出球体对象的体积;
};
```

```

sphere::sphere(double radius)
{
    r=radius;
}
void sphere::vol()
{
    //请完善该函数体，计算并输出球体的体积并使用cout输出;
}
void main()
{
    sphere s1(2.5);
    s1.vol();
}

```

(2) 设计并测试一个名为 **Rectangle** 的矩形类，其属性为矩形的左下角与右上角两个点的坐标，能计算矩形的面积。

```

#include <iostream>
using namespace std;
class Rectangle
{
public:
    Rectangle (int top, int left, int bottom, int right);
    ~Rectangle () {}
    int GetTop() const { return itsTop; }
    int GetLeft() const { return itsLeft; }
    int GetBottom() const { return itsBottom; }
    int GetRight() const { return itsRight; }

    void SetTop(int top) { itsTop = top; }
    void SetLeft (int left) { itsLeft = left; }
    void SetBottom (int bottom) { itsBottom = bottom; }
    void SetRight (int right) { itsRight = right; }
    int GetArea() const;
private:
    int itsTop;
    int itsLeft;
    int itsBottom;
    int itsRight;
};
Rectangle::Rectangle(int top, int left, int bottom, int right)
{
    itsTop = top;
    itsLeft = left;
}

```

```

    itsBottom = bottom;
    itsRight = right;
}
int Rectangle::GetArea() const
{
    int Width = itsRight-itsLeft;
    int Height = itsTop - itsBottom;
    return (Width * Height);
}
int main()
{
    Rectangle MyRectangle (100, 20, 50, 80 );
    int Area = MyRectangle.GetArea();
    cout << "Area: " << Area << "\n";
    MyRectangle.SetTop (200);
    cout << "Area: " << MyRectangle.GetArea() << "\n";
    return 0;
}

```

(3) 定义一个 CPU 类, 包含等级(rank)、频率(frequency)、电压(voltage)等属性, 有两个公有成员函数 run、stop。其中, rank 为枚举类型 CPU\_Rank, 定义为 enum CPU\_Rank{P1=1,P2,P3,P4,P5,P6,P7}, frequency 为单位是 MHz 的整型数, voltage 为浮点型的电压值。

实验步骤:

①首先定义枚举类型 CPU\_Rank, 例如 enum CPU\_Rank {P1=1,P2,P3,P4,P5,P6,P7}, 再定义 CPU 类, 包含等级(rank)、频率(frequency)、电压(voltage)等私有数据成员, 定义成员函数 run、stop, 用来输出提示信息, 在构造函数和析构函数中也可以输出提示信息。在主程序中定义一个 CPU 的对象, 调用其成员函数, 观察类对象的构造、复制构造、析构的顺序, 以及成员函数的调用。程序名: lab4\_1.cpp。

②使用 debug 调试功能观察程序 lab4\_1.cpp 的运行流程, 跟踪观察类的构造函数、析构函数、成员函数的执行顺序。参考程序如下:

```

//lab4_1.cpp
#include <iostream>
using namespace std;
enum CPU_Rank {P1=1,P2,P3,P4,P5,P6,P7};
class CPU
{
private:

```

```

CPU_Rank rank;
int frequency;
float voltage;
public:
    CPU (CPU_Rank r, int f, float v);
    CPU (CPU &object1);
    ~CPU() {cout<<"析构了一个CPU!" <<endl;}
    CPU_Rank GetRank ( ) const {return rank;}
    int GetFrequency ( ) const {return frequency;}
    float GetVoltage ( ) const { return voltage;}
    void ShowInformation();

    void SetRank (CPU_Rank r) {rank=r;}
    void SetFrequency (int f) {frequency=f;}
    void SetVoltage (float v) {voltage=v;}

    void Run ( ) {cout <<"CPU 开始运行！" <<endl;}
    void Stop ( ) {cout <<"CPU 停止运行！" <<endl;}
};

CPU::CPU (CPU_Rank r, int f, float v)
{
    rank=r;
    frequency=f;
    voltage=v;
    cout<<"构造了一个CPU! 首地址: " <<endl;
}

CPU::CPU (CPU & object1)
{
    rank=CPU_Rank(object1.rank+1);
    frequency= object1. frequency;
    voltage= object1. voltage+2;
    cout<<"复制了一个CPU!" <<endl;
}

void CPU::ShowInformation()
{
    cout<<"rank: " <<GetRank () <<endl;
    cout<<"frequency: " <<GetFrequency() <<endl;
    cout<<"voltage: " <<GetVoltage () <<endl;
    Run ( );
    Stop ( );
}

void main ( )
{

```



```
CPU a (P3,300,2.8);  
cout<<"*****CPU a 首地址: "<<&a<<"*****"<<endl;  
a.ShowInformation ();  
CPU b(a);  
cout<<"*****CPU b 首地址: "<<&b<<"*****"<<endl;  
b.ShowInformation ();  
b.SetFrequency (500);  
cout<<"b frequency: "<<b.GetFrequency() <<endl;  
cout<<"a frequency: "<<a.GetFrequency() <<endl;  
}
```

(4) 设计一个用于人事管理的 **People** (人员) 类。考虑到通用性，这里只抽象出所有类型 人员都具有的属性：**number** (编号)、**sex** (性别)、**birthday** (出生日期)、**id** (身份证号) 等。

其中"出生日期"定义为一个"日期"类内嵌子对象。

用成员函数实现对人员信息 的录入和显示。

要求包括：构造函数和析构函数、拷贝构造函数、内联成员函数、带缺省形参值的成员函数 。