



实验UNIT 05

数据的共享与保护

《程序设计》课程组



第5讲上机练习

实验目的：

1. 观察程序运行中变量的作用域、生存期和可见性；
2. 学习类的静态成员的使用；
3. 学习多文件结构在C++程序中的使用。



第5讲上机练习

实验任务：

1. 课堂练习：类的静态成员
2. 编程练习：变量的作用域和生存期，类的声明和实现，友元的定义和使用



第5讲上机练习

◆ 实验步骤提示：

1. 为每个题目建立一个新的控制台项目文件；
2. 向其中提交一个类声明的头文件，一个包含类实现的代码C++源文件；
3. 向其中提交测试类的主函数及其代码；
4. 录入代码，检查是否有错误？有则改之；
5. 选择菜单“生成解决方案”编译源程序；
6. 执行程序，观察输出结果是否正确观察输出结果是否正确，如果有错误，可以执行第6步；
7. 使用debug功能：跟踪观察类的构造函数、析构函数、成员函数的执行顺序



现在开始课堂练习！

练习内容：类的静态成员



练习1：静态数据成员

// 静态数据成员的定义和使用，请说明程序的运行结果。

```
#include <iostream>
using namespace std;
class Ctype
{
    private:
        int a;
        static int s;    //定义私有的静态数据成员s
    public:
        void Print( );
        Ctype( int x=0 );
};
```

练习1: 静态数据成员

```
void Ctype::Print( )  
{  
    cout<<"a="<<++a<<endl;  
    cout<<"s="<<++s<<endl;  
}
```

```
Ctype::Ctype( int x=0 )  
{  
    a=x;  
}
```

```
int Ctype::s=0;
```

//静态数据成员必须在类外初始化，前面不能加static

练习1：静态数据成员

```
int main()  
{  
    Ctype c1,c2,c3;  
    c1.Print();  
    c2.Print();  
    c3.Print();  
    return 0;  
}
```

请分析指出程序的运行结果，并上机验证！

静态数据成员是类的公共数据成员，对象的共享数据项

C++支持静态数据成员是为了不必使用全局变量



练习2：静态数据成员

// 静态数据成员和一般数据成员的区别

```
#include <iostream>
```

```
using namespace std;
```

```
class Student
```

```
{
```

```
    private:
```

```
        static int count;    //统计学生的总数
```

```
        int StudentNo;    //普通数据成员，学号
```

```
    public:
```

```
        Student()
```

```
        { ++count;
```

```
          StudentNo=count; }
```

练习2: 静态数据成员

```
void Print( )
{
    cout<<"Student"<<StudentNo<<" ";
    cout<<"count="<<count<<endl;
}

};

int Student::count=0;
    //静态数据成员必须在类外初始化, 前面不能加static

int main( )
{
    Student Student1;
    Student1.Print();
    count<<"-----"<<endl;
```

练习2：静态数据成员

```
Student Student2;  
Student1.Print();  
Student2.Print();  
count<<"-----"<<endl;  
Student Student3;  
Student1.print();  
Student2.Print(); Student3.Print();  
count<<"-----"<<endl;  
Student Student3;  
Student1.print(); Student2.Print();  
Student3.Print(); Student4.Print();  
return 0;  
}
```

请分析指出程序的运行结果，并上机验证！



练习3： 静态函数成员

//找出本程序中的错误!

```
#include <iostream>
using namespace std;
class Ctype
{
    private:
        int a;
        static int s;    //定义私有的静态数据成员s
    public:
        static void Print( );
        Ctype( );
};
```

练习3：静态成员函数

```
void Ctype::Print( )
{
    cout<<"a="<<++a<<endl;
    cout<<"s="<<++s<<endl; //正确
}
Ctype::Ctype( )
{
    a=0;
    s++;
    cout<<"a="<<++a<<endl;
}
int Ctype::s=0;
```

练习3：静态成员函数

```
int main()  
{  
    CType::Print();  
        //未定义对象时，直接通过类名调用静态成员函数  
    CType c1,c2;  
    c1.Print();    //可以用对象调用静态成员函数  
    c2.Print();  
    CType c3;  
    c3.Print();  
    return 0;  
}
```

请仔细阅读本程序，找出其中的错误！



本次课堂练习结束！



第5讲上机任务2

编程练习：

5-3、下面程序的运行结果是什么，实际运行一下，看看和你的设想有何不同？

```
#include <iostream>
using namespace std;
int x=5,y=7;
void myFunction{
    int y=10;
    cout<<"x from myFunction"<<x<<"\n";
    cout<<"y from myFunction"<<y<<"\n\n";
}
int main(){
    cout<<"x from main"<<x<<"\n";
    cout<<"y from main"<<y<<"\n";
    myFunction();
    cout<<"Back from myFunction!\n\n";
    cout<<"x from main"<<x<<"\n";
    cout<<"y from main"<<y<<"\n";
    return 0;
```



第5讲上机任务2

编程练习：

5-7、定义一个Cat类，拥有静态数据成员numOfCats，记录Cat的个体数目；静态成员函数getNumOfCats()，读取numOfCats。设计程序测试这个类，体会静态数据成员和静态成员函数的用法。

5-13、定义类X、Y、Z，函数h(X*)，满足：类X有私有成员i，Y的成员函数g(X*)是X的友元函数，实现对X的成员i加1；类Z是类X是友元类，其成员函数f(X*)实现对X的成员i加5；函数h(X*)是X的友元函数，实现对X的成员i加10。在一个文件中定义和实现类，在另一个文件中实现main()函数。



本讲结束



有问题吗?

