第三次非编程作业 2017312297 李姗珊 金融实验17

5-1作用域：一个标识符在程序正文中有效的区域 作用域的类型：1.函数原型作用域 2.局部作用域 3.类作用域 4.命名空间作用域

5-2 可见性：若某标识符在程序某一点上能够被引用，就说它在该处可见

一般规则：标识符要声明在前，引用在后；在同一作用域内，不能声明同名的标识符。对于在不同的作用域生命的标识符，遵循的原则是：若有两个或多个具有包含关系的作用域，外层声明的标识符如果在内层没有声明同名标识符时仍可见，如果内层声明了同名标识符时则外层标识符不可见。

5-5静态数据成员 类的静态数据成员时类的数据成员的一种特例，采用static关键字来声明，对于类的普通数据成员，每一个类的对象都拥有一份存储，就是说每个对象的同名数据成员可以分别存储不同的数值，这也是保证对象拥有自身区别于其他对象的特征的需要，但是静态数据成员，每个类只要一份存储，由所有该类的对象共同维护和使用，也就实现了同一类的不同对象之间的数据共享。

5-6静态函数成员 使用static关键字声明的函数成员是静态的，静态函数成员属于整个类，同一个类的所有对象共同维护，为这些对象所共享。静态函数成员具有以下两个方面的好处，一是由于静态成员函数只能直接访问同一个类的静态数据成员，可以保证不会对该类的其余数据成员造成负面影响；二是同一个类只维护一个静态函数成员的拷贝，节约了系统的开销，提高程序的运行效率。

5-9 B不是A的友元 C不是A的友元 D不是B的友元

5-10静态成员变量可以是私有的，

声明一个私有的静态整型成员变量

private：

static int x；

5-16 编译的输入文件是源文件，输出是目标文件，连接的输入文件是目标文件，输出是可执行文件

编译器对源代码进行编译，是将以文本形式存在的源代码翻译为机器语言形式的目标文件的过程。连接是将各个编译单元的目标文件和运行库当中被调用过的单元加以合并后生成的可执行文件的过程。

（1）编译时报错，函数参数不匹配

（2）连接错误，函数为定义

（3）不报错

（4）连接错误，函数重复定义

6-6 \*：指针运算符，表示指针所指向的对象的值

&：取地址运算符，是用来得到一个对象的地址

6-7 指针：一种数据类型，用于储存内存单元地址的变量

指针储存的地址和这个地址中的值有什么区别？ 指针存放的是另外一个对象的地址，地址中的值就是这个对象的内容

6-9 ‘\0’

6-11 引用是一个别名，不能为NULL值，不能被重新分配；指针是一个存放地址的变量。当需要对变量重新赋以另外的地址或赋值为NULL时只能使用指针。

6-15 const int \*p1是一个常量指针：p1的指向的对象的值不可变，但p1储存的地址可变（指向的对象个体可以变）

Int \* const p2是一个指针常量：p1指向的对象不能变化，但是对象的值能够发生变化

6-18 函数调用返回时（return）p指针的内存空间没有释放

解：

#include<iostream>

Using namespace std;

Int \*fn1( ){  
int \*p=new int(5);

Return p;

}

Int main( ){  
int \*a=fn1( );

Cout<<”the value of a is: ”<<\*a;

Delete a;

Return 0;

}