5-1

作用域指标识符在程序正文中的有效范围。

作用域分为函数原型作用域、块作用域、类作用域和文件作用域。

5-2

可见性指标识符知否可以引用。

一般规则：标识符先声明，后引用；同一作用域中不能声明同名标识符。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则外层标识符不可见。

5-5

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。静态数据成员属于整个类，由所有该类的对象共同维护和使用，实现了同一类不同对象之间的数据共享。

5-6

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，由所有该类的成员共同维护和使用。

优点：静态成员函数只能直接访问同一个类的静态数据成员，不会对其他成员产生影响；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。

5-9

不是。友元关系不具有交换性。

不是。友元关系不具有传递性。

不是。友元关系不能被继承。

5-10

可以。

Private:

Static int a;

5-16

编译：输入文件是源文件，输出文件是目标文件。

连接：输入文件是目标文件，输出文件是可执行文件。

编译是将源代码翻译成机器语言形式的目标文件；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可执行文件。

编译时报错。函数的参数表不匹配。

连接时报错。函数没有定义。

不会报错。

连接时报错。函数重复定义。

6-6

“\*”是指针运算符，表示指针所指向的对象的值。

“&”是地址运算符，用来得到一个对象的地址。

6-7

指针是一种数据类型，具有指针类型的变量称为指针变量。指针变量存放的是另外一个对象的地址，这个地址中的值就是另一个对象的内容。

6-9

‘\0’字符

6-11

引用是一个别名，指针是一个存放地址的变量。

当需要赋值为NULL或对变量重新赋以另外的地址时只能使用指针。

6-15

Const int \* p1声明的是一个指向整型常量的指针p1;int \* const p2声明了一个指针型常量，用于存放整型变量的地址，一经初始化就不能被重新赋值。

6-18

未释放给\*p分配的内存。

改正为

int \*fn1();

int main ()

{

int \*a =\*fn1();

cout <<”the value of a is: “<<\*a;

delete a;

return 0;

}

int \*fn1()

{

int \*p = new int (5);

return p;

}

6-19

long ( \*p ) (int);

long ( A::\*p) (int);