5-1 一个标识符在程序中的有效作用区域。函数原型作用域 局部作用域 类作用域 命名空间作用域

5-2 标识符的有效范围 程序运行到某一处 能够引用到的标识符 就是该处可见的标识符

规则 标识符要声明在前 引用在后 2、同一作用域内 不能声明同名标识符 没有互相包含关系的不同作用域中的同名的标识符互不影响。4、两个或者多个有包含关系的作用域中，同名标识符，外层不可见。

5-5静态数据成员具有静态生存期，属性属于整个类，不属于某个对象。通过类名对他进行访问。

5-6属性属于整个类，不属于某个对象。通过类名对他进行访问。直接访问该类的静态数据和 成员函数。可通过类名和对象名两种方式调用。访问非静态成员，只能通过对象名。

5-9 都不是，友元关系是单向且不可传递的。

5-10 private:  
static int x; //将静态成员变量x声明为私有

5-16 编译输入,i文件，输出二进制的目标文件。 链接输入目标文件，输出.exe文件。

（1）（2），会报错，编译阶段，(3)不会，（4）会，连接阶段。

6-6，\*指针运算符，获取指针所指向地址的值，一元运算符。&取地址，得到一个对象的地址。一元操作符。

6-7 指针类型是专门用来存放内存单元地址的变量类型

地址是内存空间，指针所指向的值是数据。

6-9 \0

6-11；引用一经初始化，无法更改被引用对象，如有这种需求，必须用指针；

没有空引用，但有空指针，如果空指针有存在的必要，必须用指针；

函数指针；

用new动态创建的对象或数组，用指针存储其地址最自然；

函数调用时，以数组形式传递大量数据时，需要用指针作为参数。

引用在底层通过指针来实现

一个引用变量，通过存储被引用对象的地址，来标识它所引用的对象

引用是对指针的包装，比指针更高级

指针是C语言就有的底层概念，使用起来很灵活，但用不好容易出错

引用隐藏了指针的“地址”概念，不能直接对地址操作，比指针更安全

6-15 前者指向整型常量的指针，指向的值不能修改。后者指向整型的常量指针，不能指向别的变量，但是指向的值可以改变。

6-18 分配动态内存后缺少delete，在下一行加delete p；

6-19 long (\* p\_fn1)(int);

long ( A::\*p\_fn2)(int);