5-1

作用域是一个标识符在程序正文中的有效区域。C++的作用域分为函数原型作用域、局部作用域、类作用域、文件作用域。

5-2

可见性是标识符是否可以引用的问题。规则：标识符声明在前，引用在后；在同一作用域不能声明同名的标识符。对于不同作用域声明的标识符，应遵循：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则仍可见，若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，采用static关键字来声明。对于类的普通数据成员，每个类的对象都有一份存储，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，由此实现了同一类的不同对象之间的数据共享。

5-6

用static声明的成员函数，静态成员函数属于整个类，由同一个类的所有对象共同维护，被所有对象共享。特点：只能访问同一类的静态数据成员，保证不会对该类其他数据成员造成负面影响；同一个类只需维护一个静态函数成员的拷贝，节约系统开销，提高程序运行效率。

5-9

类B不是类A的友元，类C不是类A的友元，类D不是类B的友元。

5-10

静态成员变量可以为私有的，

private：

static int i;

5-16

编译输入的是源文件，输出目标文件；连接输入目标文件，输出可执行文件。编译：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可执行文件的过程。

1. 编译时报错，函数参数不匹配
2. 连接时报错，函数未定义
3. 不报错
4. 连接报错，函数重复定义

6-6

\*为指针运算符，一元操作符，表示指针所指向的对象的值；&取地址运算符，一元运算符，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成为指针变量。指针变量存放的是另一个对象的地址，这个地址中的值就是另一个对象的内容。

6-9

结束符是‘\0’

6-11

语言形式上存在差异。引用是一个别名，不能为NULL，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另外的地址或赋值时只能使用指针。

6-15

const int\*p1声明了一个指向整型常量的指针p1，不能通过指针p1来改变它所指向的整型值；int \*const p2声明了一个指针型常量，用于存放整型变量的地址，这个指针一旦初始化后，就不能被重新赋值。

6-18

给\*p分配的内存未被释放

#include<iostream>

using namespace std;

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 (\*function)(int);

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

6-21

6-22