



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

设计函数 - 接口与实现分离

- ◆ 3、多模块下的实体设计
- ◆ 4、接口与实现分离

4.9 变量修饰小结

- ▶ (1) 单个文件单个函数的程序。
- ▶ 在前面章节中，并没有感受到变量修饰的复杂性，原因是我们始终在单个文件单个函数（main函数）中编写程序。尽管编写十几、几十甚至更多行的代码，但变量的应用是不复杂的。

4.9 变量修饰小结

- 变量先定义后使用（作用域规则）
- 变量使用前应该赋初值，否则为随机值（初始化规则）
- 变量定义不能同名（作用域规则）
- 变量可以在复合语句及嵌套中定义（作用域规则）
- 变量在复合语句及嵌套中定义允许同名（作用域规则）

4.9 变量修饰小结

- ▶ (2) 单个文件多个函数的程序。
- ▶ 从本节开始自定义函数，程序除了main函数外，会有多个函数出现，变量的应用也多了起来。

4.9 变量修饰小结

{ 变量分局部变量和全局变量（作用域规则）
变量分动态存储和静态存储（生命期规则）
变量初始化与存储方式有关（初始化规则）

局部变量 { 自动变量，即动态局部变量（多数情况下使用）
静态局域变量（函数多次调用仍保持数据值情况下使用）
形式参数（函数间数据传递时使用）
寄存器变量（已有编译器优化工具，极少使用）

4.9 变量修饰小结

全局变量 { 函数间数据传递，不用或少用

动态存储 { 自动变量（auto进入函数分配，函数退出释放）
形式参数（进入函数分配，函数退出释放）
寄存器变量（register进入函数分配，函数退出释放）

静态存储 { 静态局部变量（static修饰）

4.9 变量修饰小结

初始化 { 设定值（已初始化的全局变量、静态局部变量，运行前一次设置）
设定值（已初始化的动态局部变量，函数调用每次重新设置）
0（未初始化的全局变量、静态局部变量，运行前一次设置）
随机值（未初始化的动态局部变量）

4.9 变量修饰小结

- ▶ (3) 多个文件多个函数的程序。
- ▶ 编译器是按文件为单位编译的，现今的编译器都有增量编译的功能，即当编译器发现某个源文件未曾改动，那么就不重新编译它，以节省编译时间，所以即使程序的函数不多，为了提高编译效率也依然要使用多个文件的工程模式。这时变量的应用情况越来越复杂。

4.9 变量修饰小结

{ 变量和函数公有使用（作用域规则，允许多个文件中使用）
变量和函数私有使用（作用域规则，只限一个文件中使用）
实体可见（可见规则）

公有使用 { 全局变量（在需要使用的文件中extern声明）
函数（在需要使用的文件中extern声明）

私有使用 { 全局变量（在需要限定的变量定义中static声明）
函数（在需要限定的函数定义中static声明）

4.9 变量修饰小结

实体可见 {

- 文件、函数、复合语句、嵌套复合语句区域逐级包含
- 包含关系中子区域在父区域不可见
- 包含关系中父区域在子区域同名不可见、不同名可见
- 同一个父区域的平行区域互不可见
- 全局实体使用extern声明在别的文件可见
- 全局实体使用static声明仅限本文件可见

4.9 变量修饰小结

- ▶ (4) 对象保护。
- ▶ `const`限定声明对象是只读的，从而保护对象不会意外修改。

4.9 变量修饰小结

- ▶ (5) 接口与实现分离技术
- ▶ 只暴露接口，而隐藏函数的实现细节。也就是说头文件里只提供要暴露的函数的声明，其他所有代码信息都不会显示出来。用接口与实现分离的技术。
- ▶ 该技术就是将实现的一个功能封装在一个块内，提供输入输出接口，外界只需要提供输入的数据，内部运算后向外界输出，这样，将有效的实现隐藏，并起到与外界的公共函数定义不冲突等益处。

4.9 变量修饰小结

- ▶ (5) 接口与实现分离技术
- ▶ 所谓接口分离原则就是指子模块不必继承并实现接口包中不需要的东西。

CP 程序设计