

09 Memory Models and Namespaces

名字空间

将代码组织和封装在一起，【目的】以避免在大型程序中名称冲突的问题

```
1 namespace my_namespace1 {
2     // 在这里声明和定义变量、函数和类等
3     //不用担心命名冲突
4     void my_funtion(){}
5 }
6
7 namespace my_namespace2 {
8     // 在这里声明和定义变量、函数和类等
9     void my_funtion(){}
10 }
11 //使用时
12 my_namespace1::my_funtion();
13 my_namespace2::my_funtion();
```

作用域

C++中的作用域是指一个变量、函数或其他实体的可见性【即起作用的范围】。根据作用域的不同，可以将它们分为以下几类：

局部作用域 (Local Scope)

局部作用域通常是指在函数内部定义的变量或实体。这些变量仅在函数内部可见，函数执行结束后，它们将被销毁。

代码块{}：局部。函数原型，自动变量

循环体内定义的变量。这些变量仅在循环体内可见，循环结束后会被销毁。可以看作是局部作用域

```

1 void my_function(int var) {
2     int local_var = 10; // 局部变量
3 }
4
5 for (auto i = 0; i < 5; ++i) {
6     // 'i'在循环体内可见
7 }
8 // 'i'在此处不可见
9
10 {
11     int i = 1;
12 }

```

全局作用域 (Global Scope)

全局作用域是指在函数之外定义的变量或实体，它们在整个程序中都是可见的。全局变量的生命周期从程序开始执行时创建，直到程序结束时销毁。

```

1 int global_var = 20; // 全局变量
2
3 void my_function() {
4     // 在此处可以访问全局变量
5 }

```

存储持续性

指一个变量在内存中存在的时间【生命周期】。

自动存储期 (Automatic Storage Duration)

函数中声明的参数和变量；代码块中的变量

和栈关联

```

1 void my_function() {
2     int auto_var = 10; // 自动存储期变量
3
4 } //运行完函数，变量就被撤销

```

静态存储期 (Static Storage Duration)

```
1 int global_var = 20; // 静态存储期变量
2
3 void my_function() {
4     static int static_var = 30; // 静态存储期变量
5 }
```

线程存储期 (Thread Storage Duration)

动态存储期 (Dynamic Storage Duration)

```
1 int* dynamic_var = new int(50); // 动态存储期变量
2
3 // ... 其他操作 ...
4
5 delete dynamic_var; // 释放动态分配的内存
```

链接性

指名称如何在不同单元【不同文件，不同模块】间共享。

定义在函数外部的变量，链接性为外部，该变量简称为外部变量，它们的存储持续性为静态，作用域为整个文件。

每个外部变量只能有一次定义，定义声明 (defining declaration) 或简称为定义 (definition)，它给变量分配存储空间；

引用声明 (referencing declaration) 或简称为声明 (declaration)，它不给变量分配存储空间，只引用已有的变量。

```
1 // file01.cpp
2 extern int cats = 20; // definition because of
   initialization
3 int dogs = 22;
4 int fleas;
5
6 // file02.cpp
7 //use cats and dogs from file01.cpp
8 extern int cats; // not definitions because they
9 extern int dogs; // extern and have no
   initialization
10
11 // file98.cpp
```

```
12 // use cats, dogs, and fleas from file01.cpp
13 extern int cats;
14 extern int dogs;
15 extern int fleas;
```