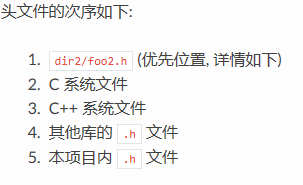
**C++编码规范： （*内容摘自Google开源项目指南，个人进行取舍编辑*）**

**头文件：**

1. 使用标准的头文件包含顺序可增强可读性



**作用域**：

1 . 只有当函数只有 10 行甚至更少时才将其定义为内联函数.

2. 将函数变量尽可能置于最小作用域内, 并在变量声明时进行初始化.

**类：**

1. 不要定义隐式类型转换.
2. 仅当只有数据成员时使用 struct, 其它一概使用 class.
3. 除少数特定环境外, 不要重载运算符. 也不要创建用户定义字面量.
4. 将 所有 数据成员声明为 private, 除非是 static const 类型成员 (遵循 [常量命名规则](https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/naming/#constant-names)).

**函数：**

1. 函数的参数顺序为: 输入参数在先, 后跟输出参数.
2. 我们倾向于编写简短, 凝练的函数.

说明：

**我们承认长函数有时是合理的, 因此并不硬性限制函数的长度. 如果函数超过 40 行, 可以思索一下能不能在不影响程序结构的前提下对其进行分割.**

**即使一个长函数现在工作的非常好, 一旦有人对其修改, 有可能出现新的问题, 甚至导致难以发现的 bug. 使函数尽量简短, 以便于他人阅读和修改代码**

1. 所有按引用传递的参数必须加上 const.
2. 若要使用函数重载, 则必须能让读者一看调用点就胸有成竹, 而不用花心思猜测调用的重载函数到底是哪一种. 这一规则也适用于构造函数.
3. 只允许在非虚函数中使用缺省参数, 且必须保证缺省参数的值始终一致. 缺省参数与 [函数重载](https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/others/#function-overloading) 遵循同样的规则. 一般情况下建议使用函数重载, 尤其是在缺省函数带来的可读性提升不能弥补下文中所提到的缺点的情况下.