**C++编码规范**

**包含文件的名称及次序**

　　将包含次序标准化可增强可读性、避免隐藏依赖（hidden dependencies，注：隐藏依赖主要是指包含的文件编译），次序如下：C 库、C++库、其他库的.h、项目内的.h。

项目内头文件应按照项目源代码目录树结构排列，并且避免使用UNIX文件路径.（当前目录）和..（父目录）。

类的编码规范：

1. 仅当只有数据时使用struct，其它一概使用class

2、当子类继承基类时，子类包含了父基类所有数据及操作的定义。 C++实践中，继承主要用于两种场合： 实现继承（ implementation inheritance） ，子类继承父类的实现代码； 接口继承（ interface inheritance） ，子类仅继承父类的方法名称

命名方面规范：

1、通用命名规则（ General Naming Rules）  
函数命名、变量命名、文件命名应具有描述性，不要过度缩写，类型和变量应该是名词， 函  
数名可以用“命令性”动词

文件命名（ File Names）文件名要全部小写，可以包含下划线（ \_）或短线（ -），按项目约定来

类型命名（ Type Names）  
类型命名每个单词以大写字母开头，不包含下划线： MyExcitingClass、 MyExcitingEnum。  
所有类型命名——类、结构体、类型定义（ typedef）、枚举——使用相同约定

4. 变量命名（ Variable Names）  
变量名一律小写，单词间以下划线相连，类的成员变量以下划线结尾，如  
my\_exciting\_local\_variable、 my\_exciting\_member\_variable\_

5.普通函数大小写混合，  
存取函数（ accessors and mutators） 则要求与变量名匹配： MyExcitingFunction()、  
MyExcitingMethod()、 my\_exciting\_member\_variable()、  
set\_my\_exciting\_member\_variable()。  
普通函数：  
函数名以大写字母开头，每个单词首字母大写，没有下划线：  
AddTableEntry()  
DeleteUrl()  
存取函数：  
存取函数要与存取的变量名匹配，这儿摘录一个拥有实例变量num\_entries\_的类：  
class MyClass {  
public:  
...  
int num\_entries() const { return num\_entries\_; }  
void set\_num\_entries(int num\_entries) { num\_entries\_ = num\_entries; }  
private:  
int num\_entries\_;  
};  
其他短小的内联函数名也可以使用小写字母， 例如， 在循环中调用这样的函数甚至都不需要  
缓存其值，小写命名就可以接受

代码：

1. 函数名之后不要留空格，紧跟左括号‘ （’，以与关键字区别。
2. 象 if、 for、 \*\*while \*\*等关键字之后应留一个空格再跟左括号‘ （’，以突出关键字。

3、‘，’之后要留空格。在for语句中的‘；’其后要留空格，如 for (i=0; i<5; i++)。

1. 对于表达式比较长的 \*\*for \*\*语句和 \*\*if \*\*语句，为了紧凑起见可以适当地去掉一些空格。

5、赋值操作符、比较操作符、算术操作符、逻辑操作符、位域操作符，如“ =”、“ +=” “>=”、“ <=”、“ +”、“ \*”、“ %”、“ &&”、“ <<”、 “ ^”等二元操作符的前后应当加空格。

1. 一元操作符如“ !”、“ ~”、“ ++”、“ --”、“ &”（地址运算符）等前后不加空格。

7、在实现函数时，函数与函数之间应加空行。

8、在一个函数体内，逻揖上密切相关的语句之间不加空行，其它地方应加空行。

当程序中没有空行时是这样的。

9、一行代码只做一件事情。

10、长表达式要在低优先级操作符处拆分成新行，操作符放在新行之首 （以便突出操作符）。拆分出的新行要进行适当的缩进，使排版整齐，语句可读。

**格式**

　　1、行宽原则上不超过80列；

　　2、尽量不使用非ASCII字符，如果使用的话，参考 UTF-8 格式，尽量不将字符串常量耦合到代码中；

3、UNIX/Linux下无条件使用空格;

4、函数参数、逻辑条件、初始化列表：要么所有参数和函数名放在同一行，要么所有参数并排分行；

　　5、除函数定义的左大括号可以置于行首外，包括函数/类/结极体/枚举声明、各种语句的左大括号置于行尾，所有右大括号独立成行；

　　6、./->操作符前后丌留空格，\*/&不要前后都留，一个就可，靠左靠右依各人喜好；

　　7、预处理指令/命名空间不使用额外缩进，类/结构体/枚举/函数/语句使用缩进；

　　8、return不要加()；

　　9、水平/垂直留白不要滥用，怎么易读怎么来。

注释风格：

1、注释风格

用//或/\* \*/，统一就好。  
//或/\* \*/都可以， //只是用的更加广泛，在如何注释和注释风格上确保统

2、类注释（ Class Comments）  
每个类的定义要附着描述类的功能和用法的注释

3、函数注释（ Function Comments

函数声明处注释描述函数功能，定义处描述函数实现

常变量名本身足以很好说明变量用途，特定情况下，需要额外注释说明

4、类数据成员：  
每个类数据成员 （也叫实例变量或成员变量） 应注释说明用途， 如果变量可以接受NULL或-1  
等警戒值（ sentinel values） ，须说明之

全局变量（常量） ：  
和数据成员相似，所有全局变量（常量）也应注释说明含义及用途.