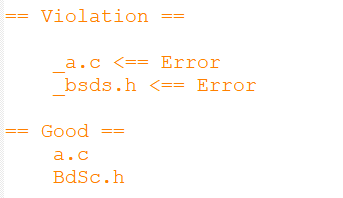
注：根据需求选择所需规则，格式如下：（~ 后有一个空格，多个规则间用英文的分号分隔）

~ RULE\_3\_1\_A\_do\_not\_start\_filename\_with\_underbar；

~ RULE\_3\_2\_B\_do\_not\_use\_same\_filename\_more\_than\_once

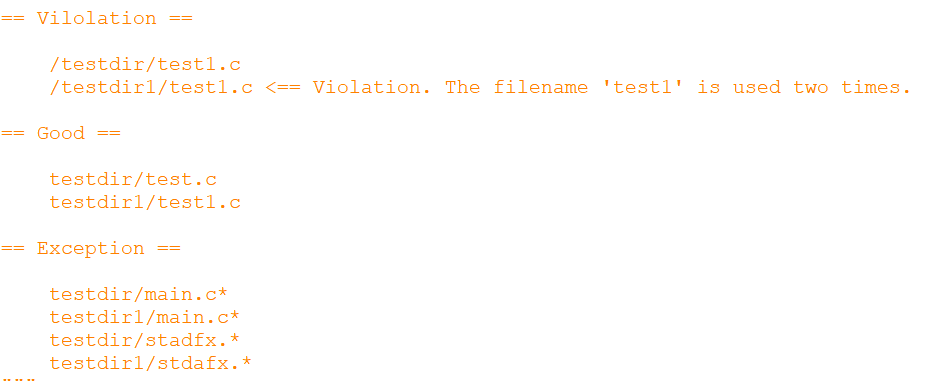
1. ~ RULE\_3\_1\_A\_do\_not\_start\_filename\_with\_underbar

文件名不能以下划线（\_）开始



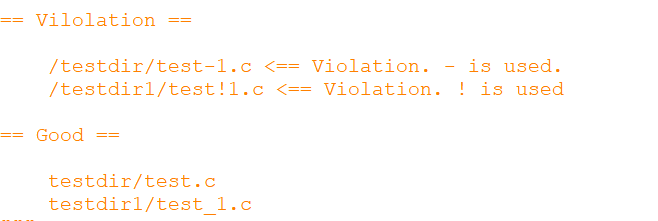
2、~ RULE\_3\_2\_B\_do\_not\_use\_same\_filename\_more\_than\_once

同一文件名不要使用超过一次



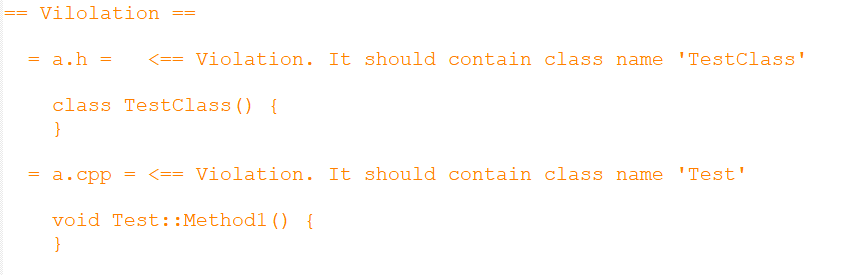
3、~ RULE\_3\_2\_CD\_do\_not\_use\_special\_characters\_in\_filename

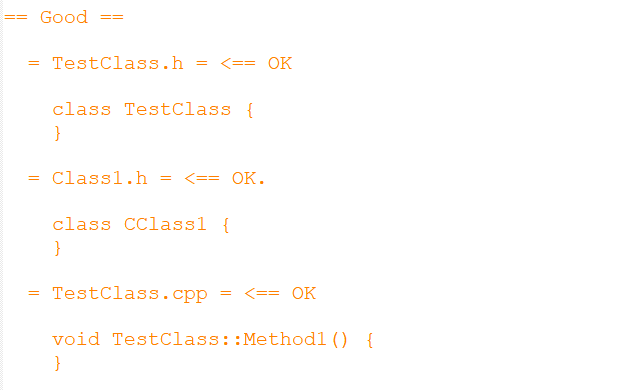
文件名中不能用特殊字符，文件名只能使用字母、数字和下划线。



4、~ RULE\_3\_2\_F\_use\_representitive\_classname\_for\_cpp\_filename

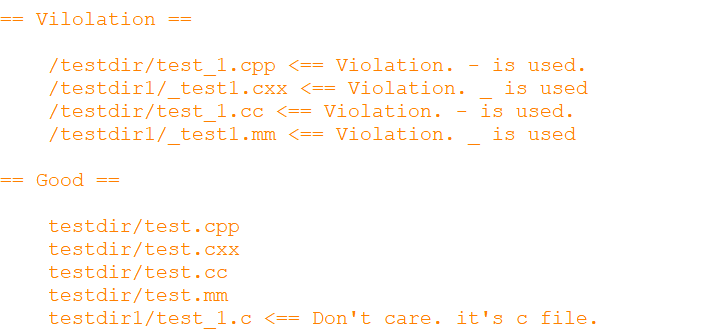
文件名要和类名一致





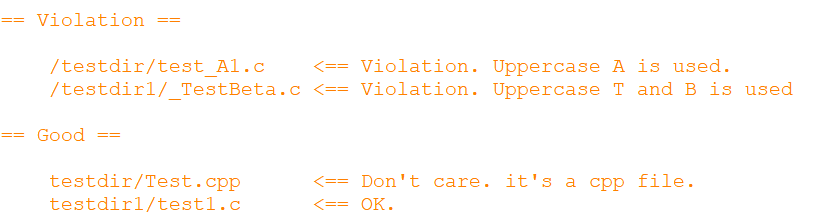
5、~ RULE\_3\_2\_H\_do\_not\_use\_underbars\_for\_cpp\_filename

文件名中不能有下划线



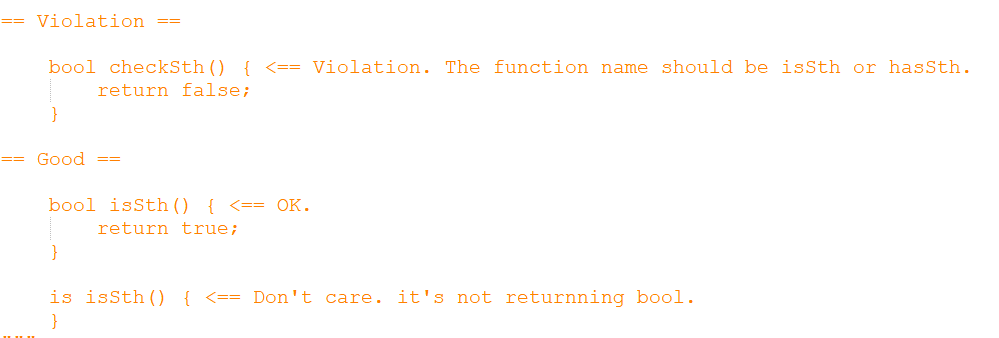
6、~ RULE\_3\_2\_H\_do\_not\_use\_uppercase\_for\_c\_filename

后缀名为.c的文件名不能使用大写字母



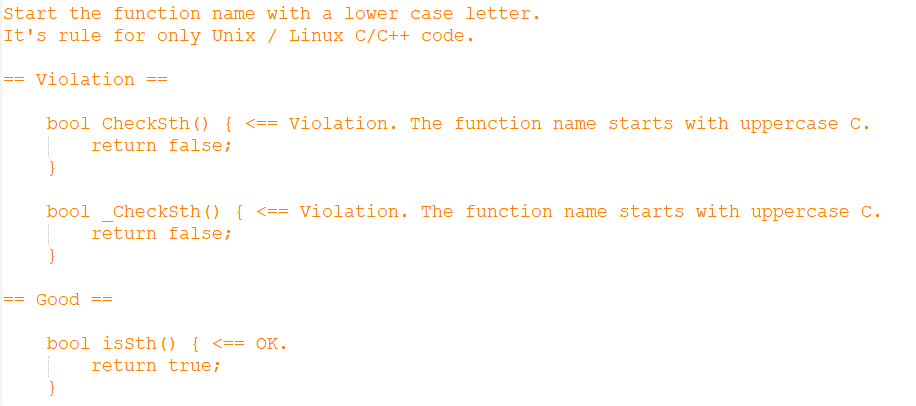
7、~ RULE\_3\_3\_A\_start\_function\_name\_with\_is\_or\_has\_when\_return\_bool

当返回值为布尔值的函数，其函数名要以is或者has开头



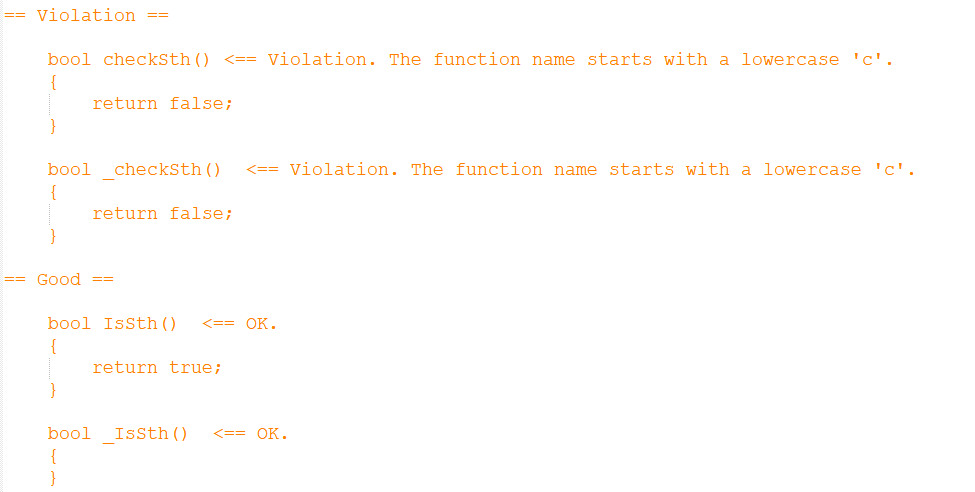
8、~ RULE\_3\_3\_A\_start\_function\_name\_with\_lowercase\_unix

unix操作系统下（实验发现windows下可以检查），函数名要以小写字母开头



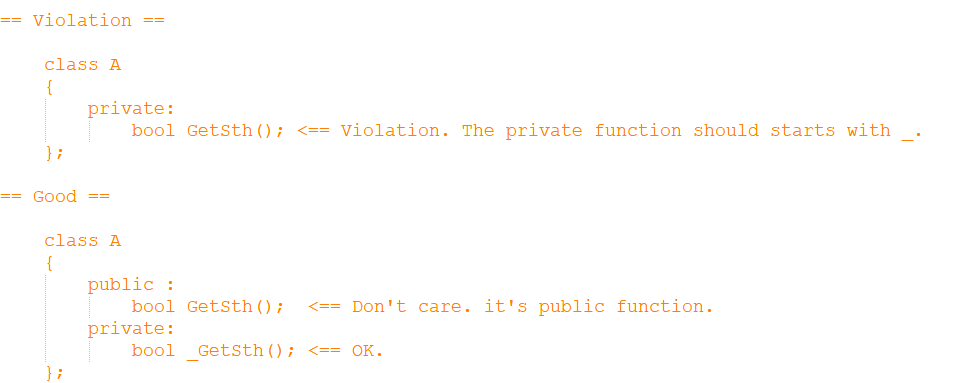
9、~ RULE\_3\_3\_A\_start\_function\_name\_with\_upperrcase\_windows

windows操作系统下，函数名要以大写字母开头



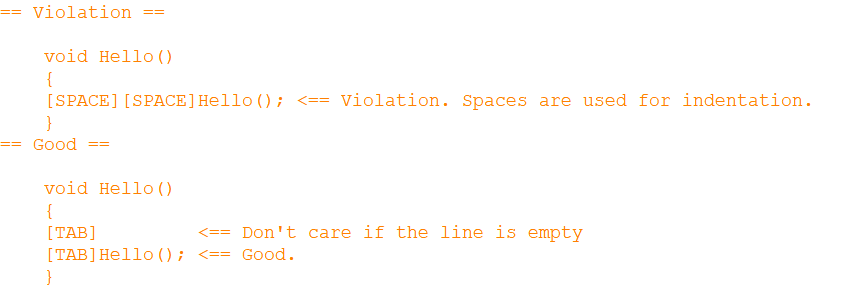
10、~ RULE\_3\_3\_B\_start\_private\_function\_name\_with\_underbar

私有函数名要以下划线开头（只使用于cpp文件）



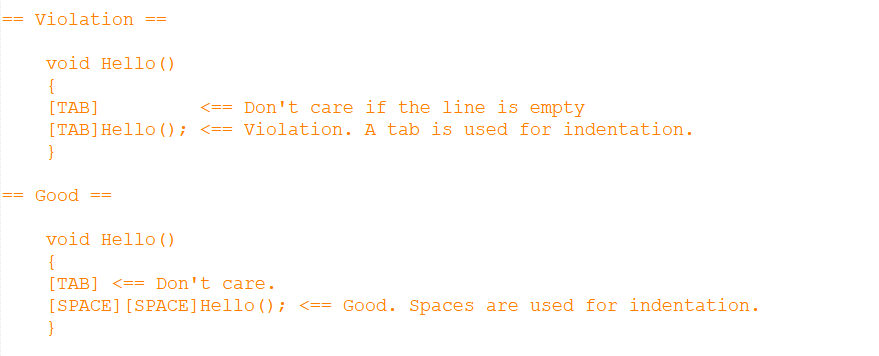
11、~ RULE\_4\_1\_A\_A\_use\_tab\_for\_indentation

用tab键作为缩进（即：代码使用缩进必须用tab键，或者代码不进行缩进。若用space键进行缩进就会报错）



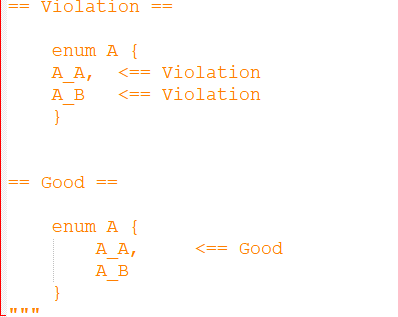
12、~ RULE\_4\_1\_A\_B\_use\_space\_for\_indentation

用space键作为缩进（即：代码使用缩进必须用space键，或者代码不进行缩进。若用tab键进行缩进就会报错）



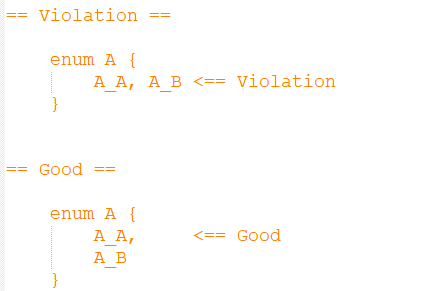
1. ~ RULE\_4\_1\_B\_indent\_each\_enum\_item\_in\_enum\_block

定义枚举类型时，每一个枚举里面的枚举项都要进行缩进



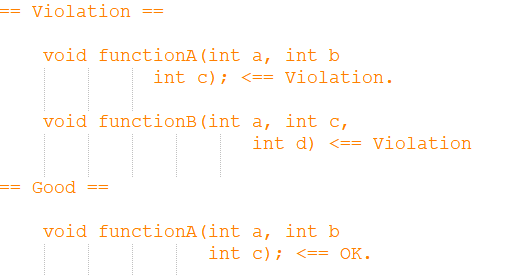
1. ~ RULE\_4\_1\_B\_locate\_each\_enum\_item\_in\_seperate\_line

定义枚举类型时，每一个单独的枚举项占一行，若一行中出现了两个或多个枚举项则会报错



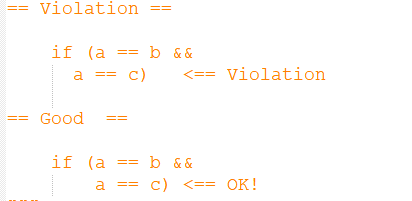
1. ~ RULE\_4\_1\_C\_align\_long\_function\_parameter\_list

当第一个函数的参数在多行中定义时，从第二行开始，每一行的第一个参数对其第一行的第一个参数。



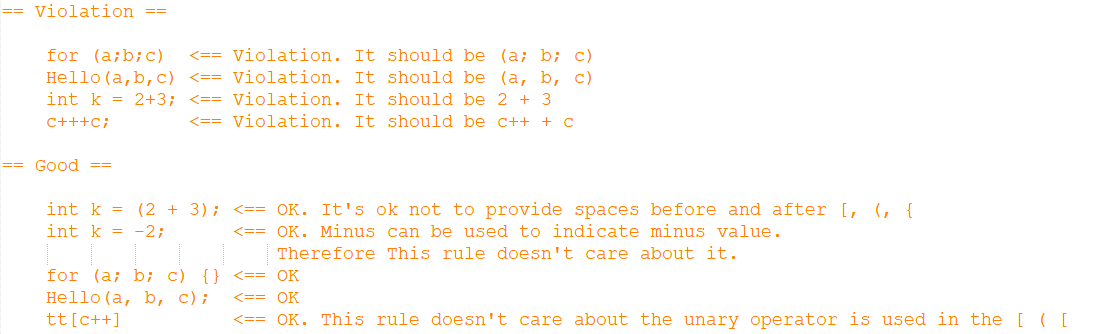
1. ~ RULE\_4\_1\_E\_align\_conditions

在if和while字句中，如果字句中的条件列表为多行，那么从第二行开始，每一个行的条件要对其第一行。



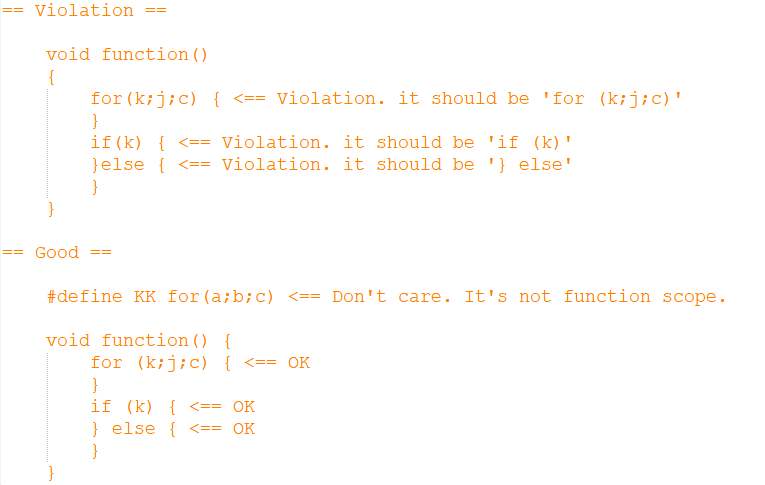
1. ~ RULE\_4\_2\_A\_A\_space\_around\_operator

操作符周围应该使用空格



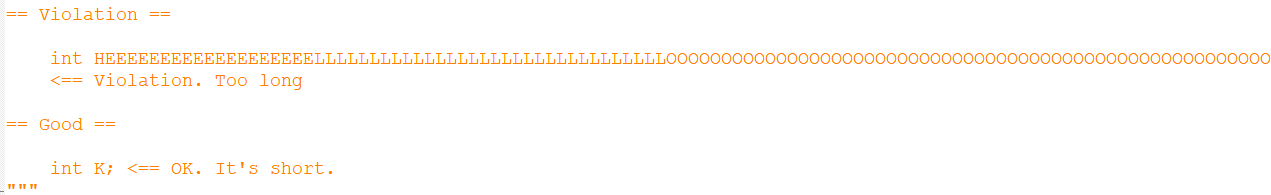
1. ~ RULE\_4\_2\_A\_B\_space\_around\_word

使用关键词if、else、for时，关键词和括号间要有间隔一个空格



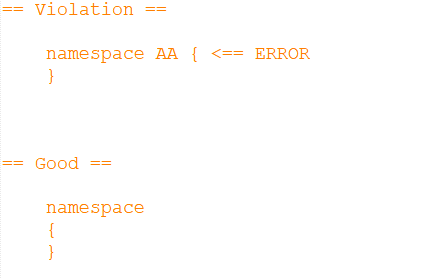
1. ~ RULE\_4\_4\_A\_do\_not\_write\_over\_120\_columns\_per\_line

每行代码不要写超过120列（这个规则不能识别制表符，它只认为每个字符是一个列）



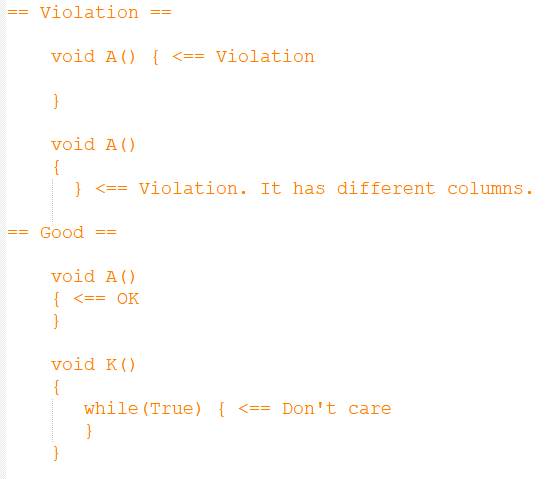
1. ~ RULE\_4\_5\_A\_brace\_for\_namespace\_should\_be\_located\_in\_seperate\_line

名称空间的大括号应该位于单独的行中。



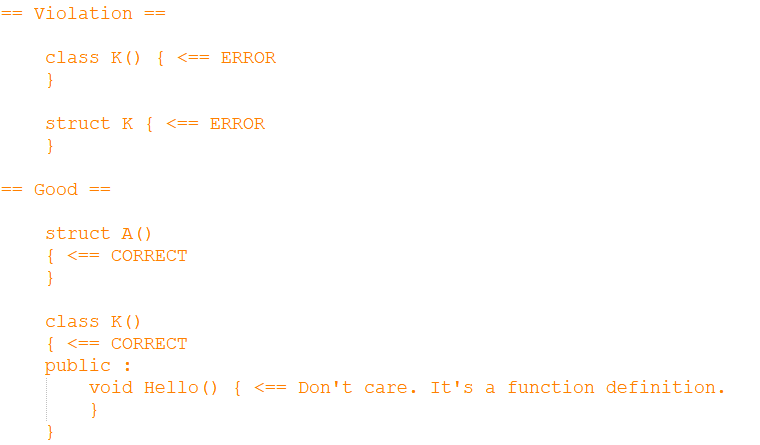
1. ~ RULE\_4\_5\_A\_braces\_for\_function\_definition\_should\_be\_located\_in\_seperate\_line

函数定义的大括号应该位于单独的行中。



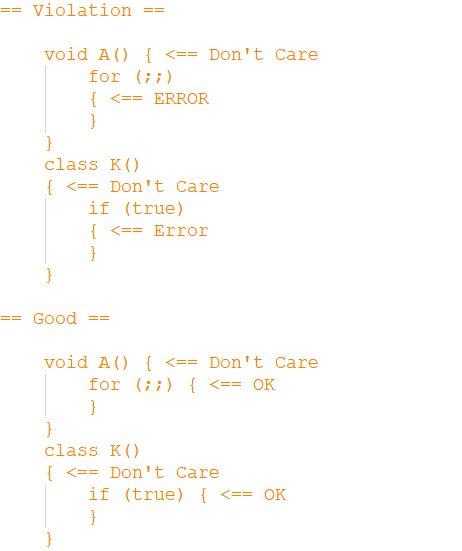
1. ~ RULE\_4\_5\_A\_braces\_for\_type\_definition\_should\_be\_located\_in\_seperate\_line

类型定义(class / struct / union / enum)的大括号应该位于单独的行中。



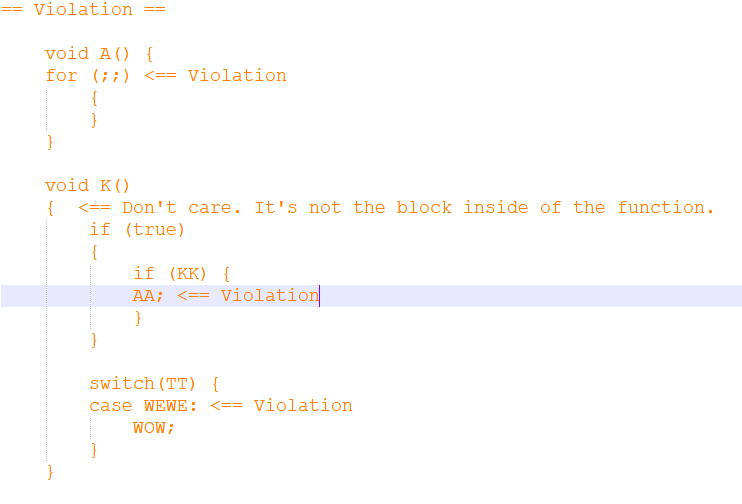
1. ~ RULE\_4\_5\_A\_braces\_inside\_of\_function\_should\_be\_located\_in\_end\_of\_line

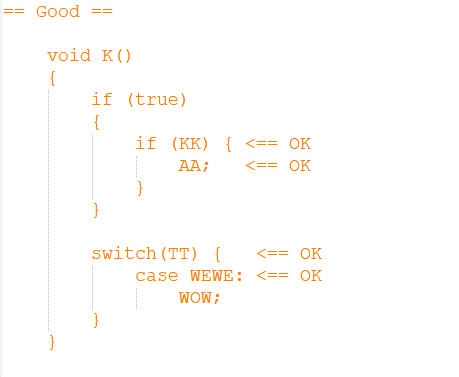
函数内部的大括号，应该位于每一行的结尾



1. ~ RULE\_4\_5\_A\_indent\_blocks\_inside\_of\_function

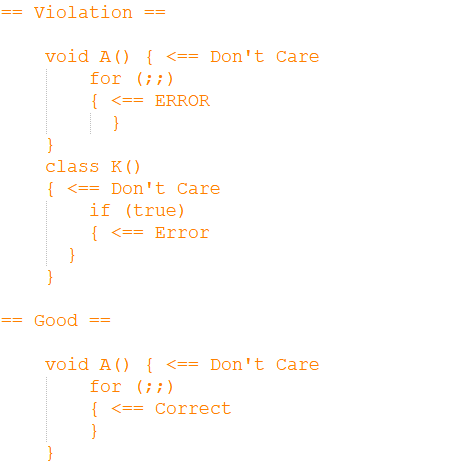
在函数内部，每一对大括号里面的代码要进行缩进，使之包含在大括号里





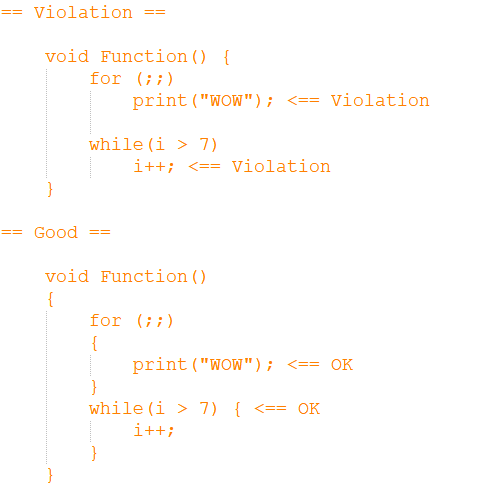
1. ~ RULE\_4\_5\_A\_matching\_braces\_inside\_of\_function\_should\_be\_located\_same\_column

在函数内部，每一对大括号应该在同一列



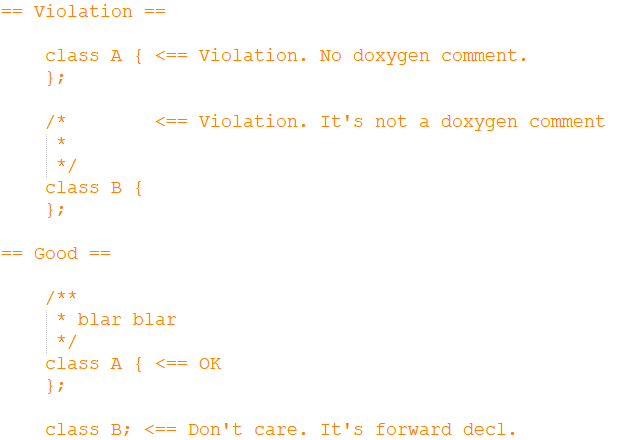
1. ~ RULE\_4\_5\_B\_use\_braces\_even\_for\_one\_statement

关键词（"IF", "WHILE", "FOR"，"ELSE"）内部只有一条语句，也要使用大括号把语句包含起来。



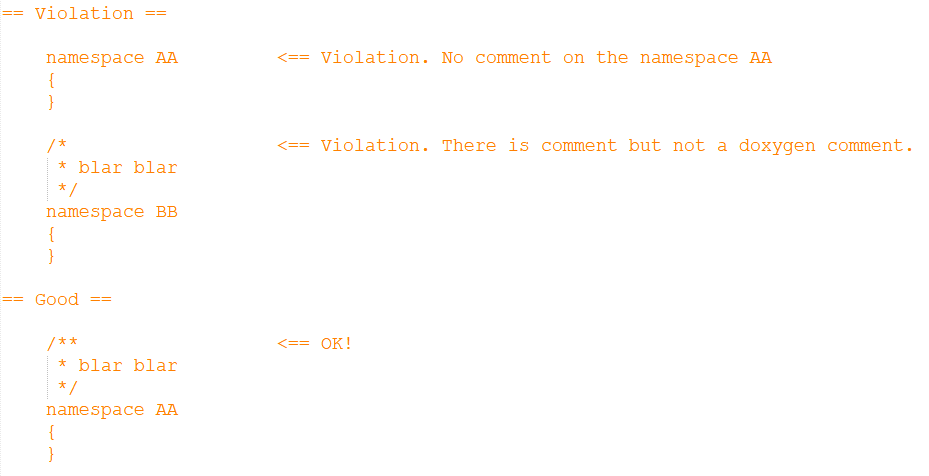
1. ~ RULE\_5\_2\_C\_provide\_doxygen\_class\_comment\_on\_class\_def

在每一个类的定义前面提供doxygen格式的注释



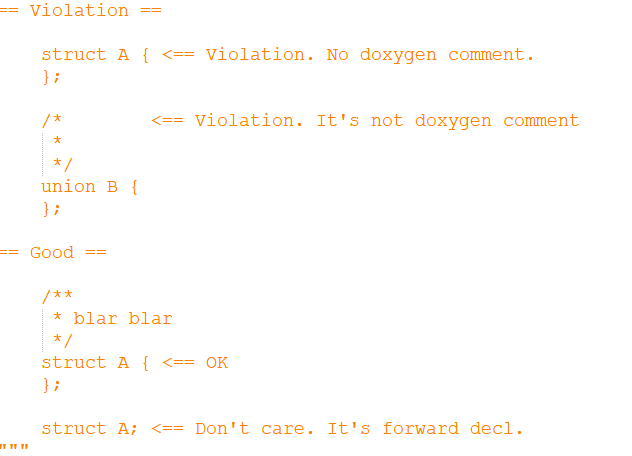
1. ~ RULE\_5\_2\_C\_provide\_doxygen\_namespace\_comment\_on\_namespace\_def

提供命名空间doxygen注释。它检查每个“名称空间”关键字前是否有doxygen sytle注释。



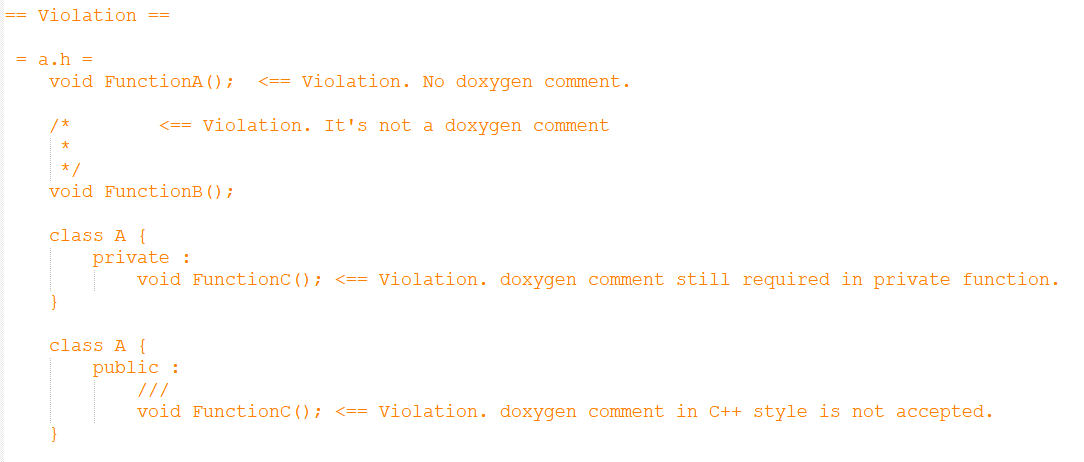
1. ~ RULE\_5\_2\_C\_provide\_doxygen\_struct\_comment\_on\_struct\_def

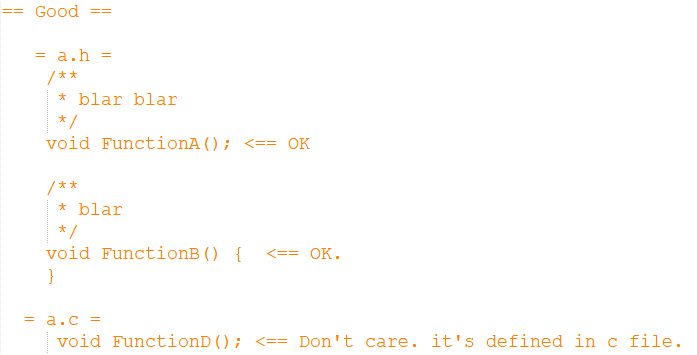
提供struct/union doxygen风格的注释。它检查每个结构体/联合体定义前面是否有doxygen sytle注释。



1. ~ RULE\_5\_3\_A\_provide\_doxygen\_function\_comment\_on\_function\_in\_header

检测头文件（.h文件）中每个函数前面是否有doxygen风格的注释（它只检查public、protected、private函数）。





1. RULE\_5\_3\_A\_provide\_doxygen\_function\_comment\_on\_function\_in\_impl

检测在impl文件（.c或.cpp文件）中是否提供doxygen注释函数。

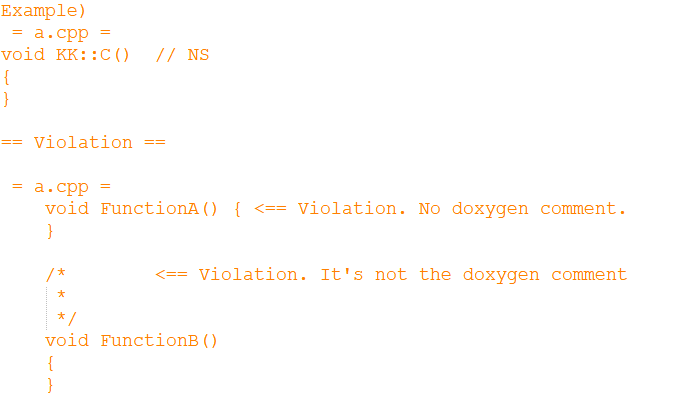
它检查每个函数定义前面是否有doxygen sytle注释。

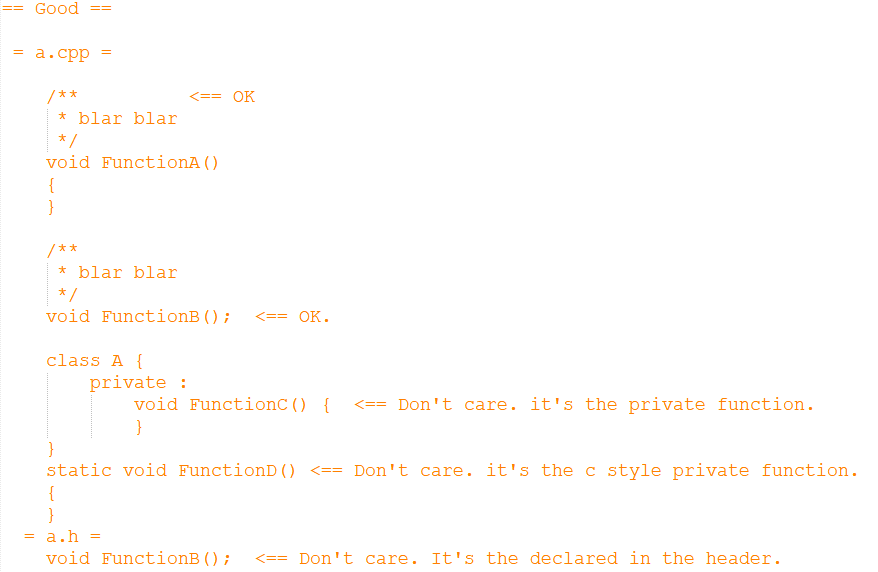
它只检查非静态和非私有函数定义。

不幸的是，这个规则不能确定方法是否是私有的，

如果函数定义位于cpp文件中。

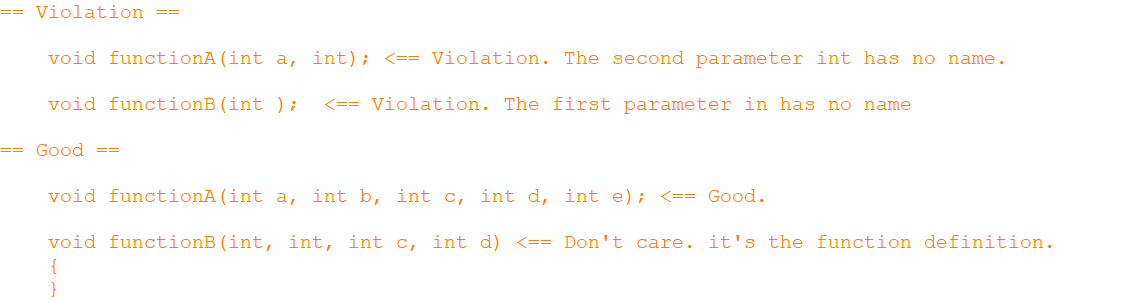
请在私隐函数签名的右侧加上'// NS'，以抑制误报。





1. ~ RULE\_6\_1\_A\_do\_not\_omit\_function\_parameter\_names

不要在函数声明中省略函数参数名。它只检查函数声明，不检查函数实现。

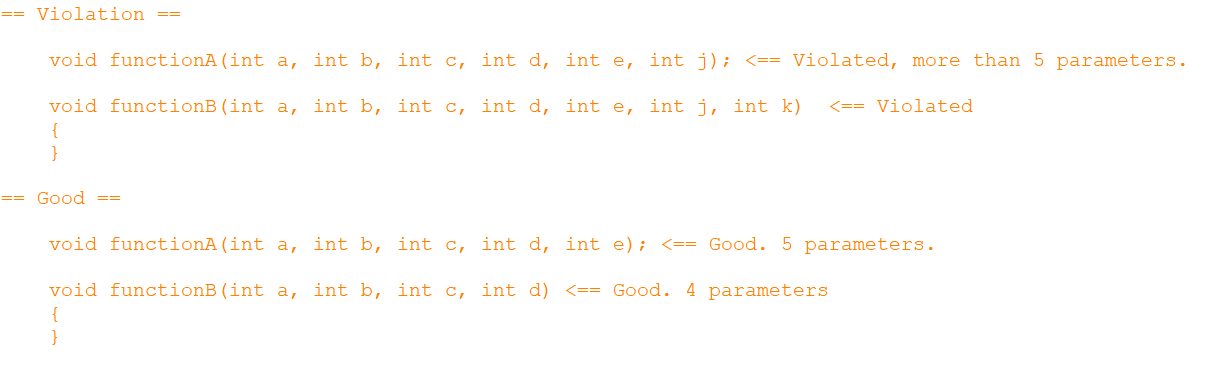


1. ~ RULE\_6\_1\_E\_do\_not\_use\_more\_than\_5\_paramters\_in\_function

每个函数中使用的参数不要超过5个。它同时检查函数声明和实现。

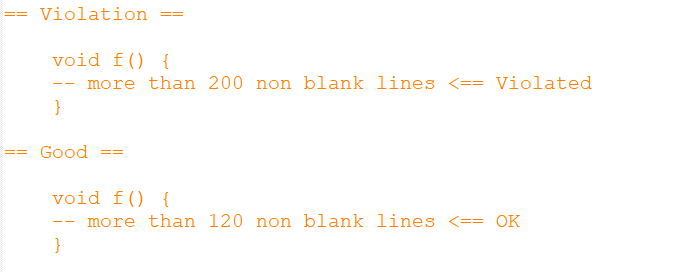
当在参数中使用多个模板(KK<T, K>)时，会混淆这条规则。

当您必须使用这样的句子时，请在行尾使用// NS来忽略这个错误。



1. ~ RULE\_6\_1\_G\_write\_less\_than\_200\_lines\_for\_function

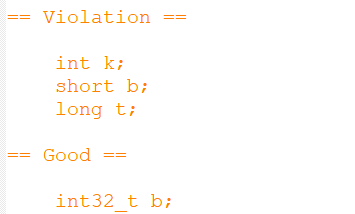
每一个函数内部语句不应该超过两百行（只检查非空行，空白行不加入计算）



1. ~ RULE\_6\_2\_A\_do\_not\_use\_system\_dependent\_type

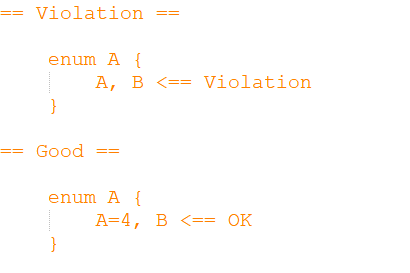
不要使用依赖于系统的类型(short、long、int)

而是分别使用与系统无关的类型(int16\_t、int64\_t、int32\_t)



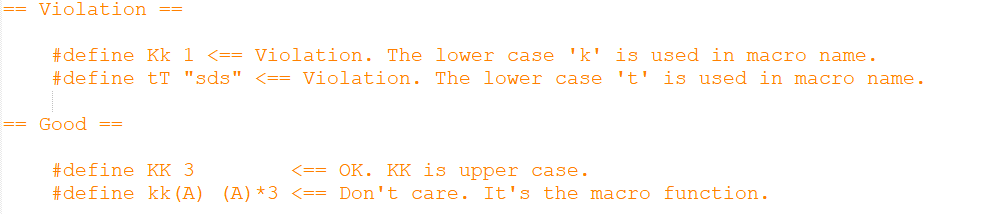
1. ~ RULE\_6\_4\_B\_initialize\_first\_item\_of\_enum

检查enum类型的第一项，并检查它是否已初始化（它检查第一个元素，并看到有= token）。



1. ~ RULE\_6\_5\_B\_do\_not\_use\_lowercase\_for\_macro\_constants

不要对宏常量使用小写字母（不过，可以使用小写字母编写宏函数）。



1. ~ RULE\_6\_5\_B\_do\_not\_use\_macro\_for\_constants

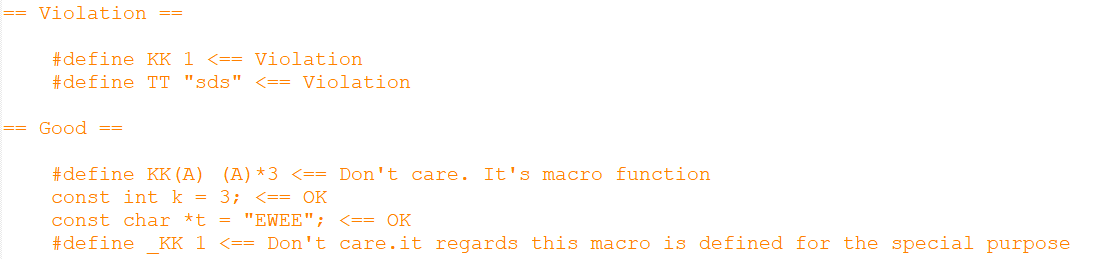
不要使用宏常量。

相反，可以使用enum或const变量。

不过，可以编写宏函数。

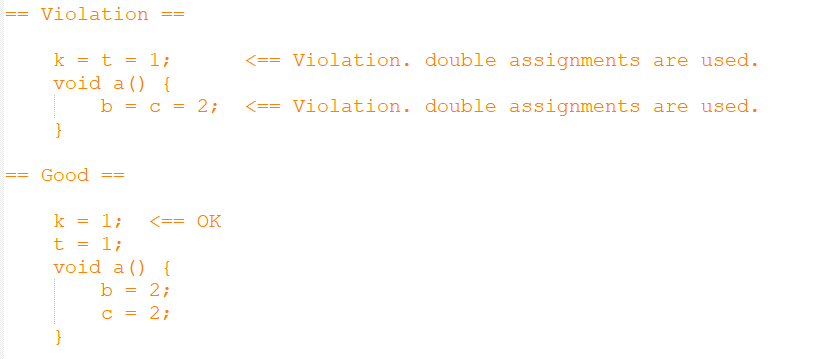
如果宏定义是以下划线开头，

它认为这个宏是为特殊目的而定义的，不会报错。



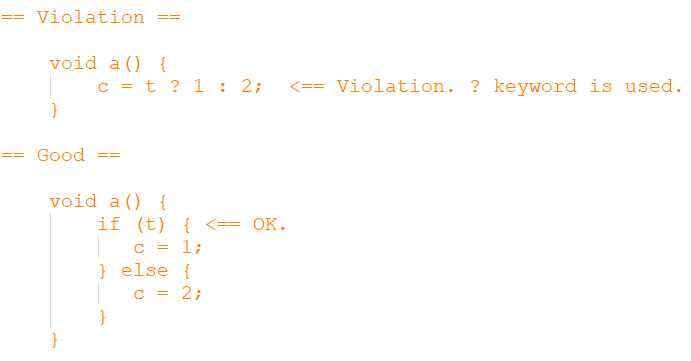
1. ~ RULE\_7\_1\_B\_A\_do\_not\_use\_double\_assignment

不用对多个变量使用双重赋值



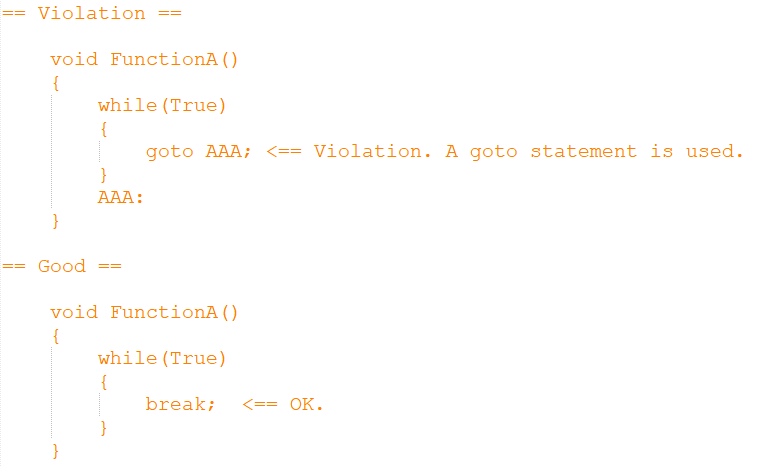
1. ~ RULE\_7\_1\_C\_do\_not\_use\_question\_keyword

代码中不要使用条件运算符”?”



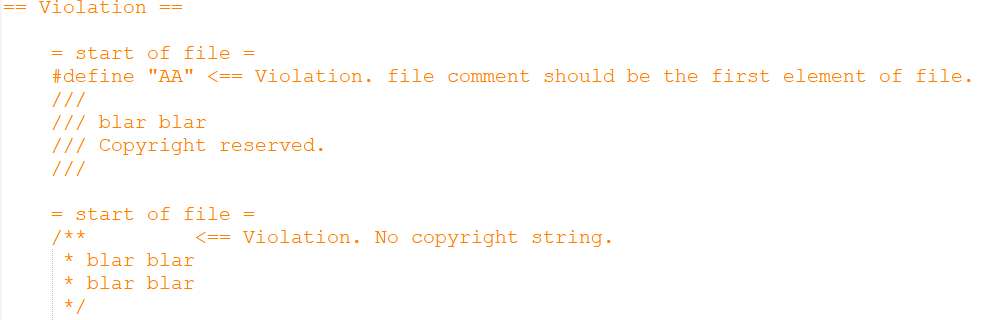
1. ~ RULE\_7\_2\_B\_do\_not\_use\_goto\_statement

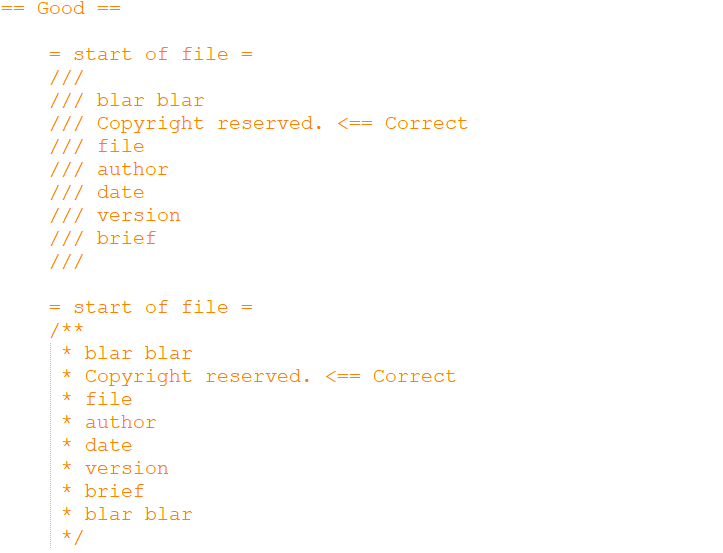
不要使用goto语句



1. ~ RULE\_8\_1\_A\_provide\_file\_info\_comment

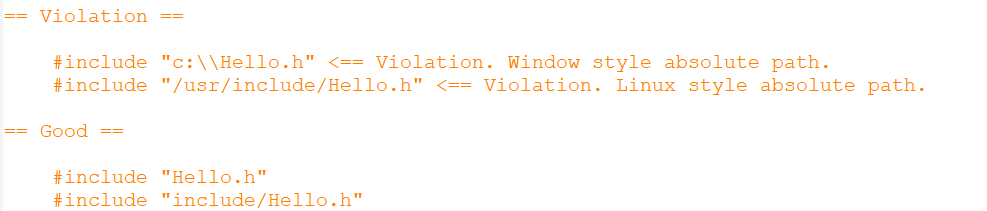
提供代码信息注释，包括版权信息，并且这些注释在代码文件的开头





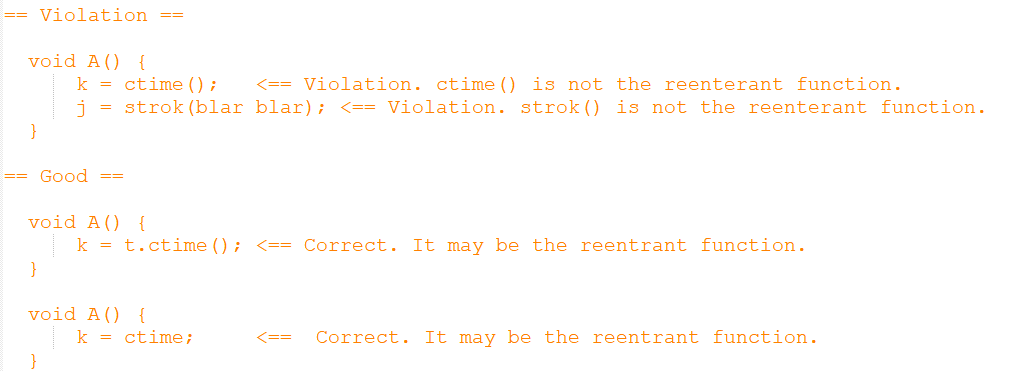
1. ~ RULE\_9\_1\_A\_do\_not\_use\_hardcorded\_include\_path

头文件路径不要使用绝对路径



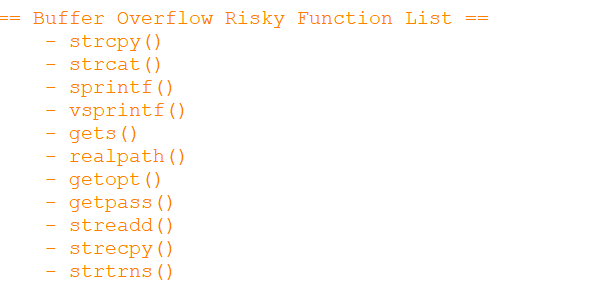
1. ~ RULE\_9\_2\_D\_use\_reentrant\_function

使用可重入函数，不要使用不可重入函数



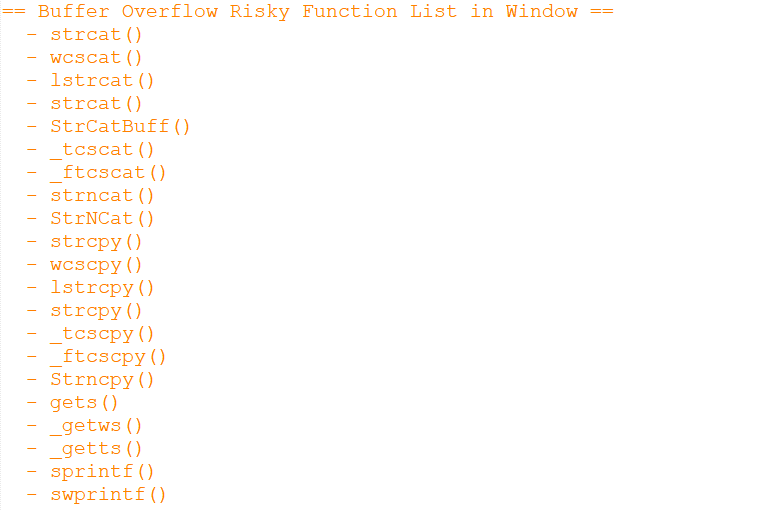
1. ~ RULE\_10\_1\_A\_do\_not\_use\_bufferoverflow\_risky\_function\_for\_unix

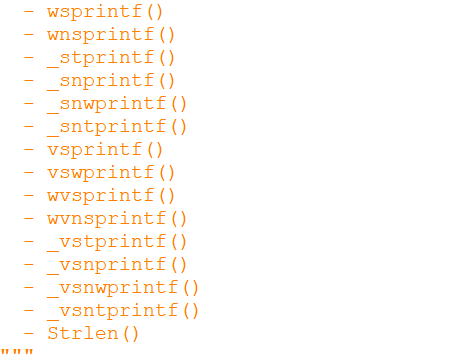
在unix环境下，不要使用缓冲区溢出危险函数



1. ~ RULE\_10\_1\_B\_do\_not\_use\_bufferoverflow\_risky\_function\_for\_windows

在windows环境下，不要使用缓冲区溢出危险函数





47、~ RULE\_A\_3\_avoid\_too\_deep\_blocks

函数中，块的深度（嵌套深度）不能大于4

