Noncopyable:

namespace utils {

class NonCopyable

{

protected:

NonCopyable() {}

~NonCopyable() {}

private:

NonCopyable(const NonCopyable &);

NonCopyable &operator=(const NonCopyable &);

};

} // end namespace utils

这里的设计思想是让子类继承，但是阻止子类调用赋值和copy构造函数（把构造函数和析构函数设置protected权限，子类可以调用，但外面的类不能调用。同时把copy构造函数和copy赋值函数做成了private，这样子类没有定义的情况下，外面的调用者不能通过赋值和copy构造等手段产生一个新的子类对象）。由于singleton模式是希望全局只有一个，这样只要让singleton直接继承noncopyalbe就行了，至少可以不用多写赋值和copy构造函数。

复制构造函数：复制构造函数的参数是**类对象本身的引用，**用于根据一个已存在的对象复制出一个新的该类的对象，一般在函数中会将因**存在对象的数据成员的值赋值一份到新创建的对象中**。（即NonCopyable(const NonCopyable &);）若没有显式写复制构造函数，系统会默认创建一个，但当类中有指针成员时，由系统默认创建该复制构造函数会存在风险。

等号运算符重载：类似复制构造函数，将=右边的本类对象的值赋值给等号左边的对象，他不属于构造函数，**等号左右两边的对象必须已经被创建**。