

CS 485
Advanced Object Oriented Design

Singleton (Ch 21)

Spring 2019

<http://www.netobjectives.com/PatternRepository/index.php?title=PatternsByAlphabet>

<http://www.netobjectives.com/files/books/dpe/design-patterns-matrix.pdf>

Singleton

- Creational Pattern
- One instance of the class with a global point of access.
 - Only ONE of something in the entire system
 - Any change to this object is automatically visible to the entire system

Singleton
- mSingleton : Singleton «static»
- Singleton()
+ getInstance() : Singleton «static»

- Often disparanged as just a global variable
 - can make code hard to read/follow
 - hard to track who uses the Singleton

Goals

- Centralized management of a resource
- Allow access from different parts of the system
- Exactly one object instantiated

Often used for....

- Logging
 - canonical example
 - `Log::Instance().logWarning("It broke");`
- Factories
 - one particular factory is used across the system
 - generate unique IDs
- Synchronized data store

Example

```
class Singleton
{
public:

    // More Effective C++: S. Meyers. (modified)

    static Singleton& Instance ();

    // non-static public methods
    void incrCount ();

    friend std::ostream& operator<<(std::ostream &rcOut,
        const Singleton &rcData);

private:
    Singleton (); // ctor is hidden
    Singleton (Singleton const&) = delete; // copy ctor is hidden

    ~Singleton (); // not virtual, no subclass. hidden so no user can delete obj.

    Singleton& operator=(Singleton const&) = delete; // assign op is hidden

    // private data members
    int mCounter = 0;
};
```

[Meyers] More Effective C++:
S.Meyers.

References

<http://www.cplusplus.com/forum/general/124047/>

<http://jrruethe.github.io/blog/2015/08/02/singletons/>

Lots of good stuff in this one, lots of bad stuff too. Can you sort through it? Hint: Look at the second reference.

<http://preshing.com/20130930/double-checked-locking-is-fixed-in-cpp11/>

C++11

- When is the Singleton object created?
- When does the Singleton object get destroyed?
- Is this implementation thread safe?
 - why might this not be thread safe?