

[Download C++ \(PDF\)](#)

Lazy Initialization

Example

This example has been lifted from the [Q & A](#) section here: <http://stackoverflow.com/a/1008289/3807729>

See this article for a simple design for a lazy evaluated with guaranteed destruction singleton:

[Can any one provide me a sample of Singleton in c++?](#)

The classic lazy evaluated and correctly destroyed singleton.

```
class S
{
public:
    static S& getInstance()
    {
        static S    instance; // Guaranteed to be destroyed.
                                // Instantiated on first use.

        return instance;
    }
private:
    S() {};                    // Constructor? (the {} brackets) are needed here.

    // C++ 03
    // =====
    // Dont forget to declare these two. You want to make sure they
    // are unacceptable otherwise you may accidentally get copies of
    // your singleton appearing.
    S(S const&);               // Don't Implement
    void operator=(S const&); // Don't implement

    // C++ 11
    // =====
    // We can use the better technique of deleting the methods
    // we don't want.
public:
    S(S const&)                = delete;
    void operator=(S const&)  = delete;

    // Note: Scott Meyers mentions in his Effective Modern
    //       C++ book, that deleted functions should generally
    //       be public as it results in better error messages
    //       due to the compilers behavior to check accessibility
    //       before deleted status
};
```

See this article about when to use a singleton: (not often)

[Singleton: How should it be used](#)

See this two article about initialization order and how to cope:

[Static variables initialisation order](#)

[Finding C++ static initialization order problems](#)

See this article describing lifetimes:

[What is the lifetime of a static variable in a C++ function?](#)

See this article that discusses some threading implications to singletons:

[Singleton instance declared as static variable of GetInstance method](#)

See this article that explains why double checked locking will not work on C++:
[What are all the common undefined behaviours that a C++ programmer should know about?](#)



PDF - Download C++ for free

Facebook Twitter More

Previous

Next

This modified text is an extract of the original [Stack Overflow Documentation](#) created by following [contributors](#) and released under [CC BY-SA 3.0](#)
This website is not affiliated with [Stack Overflow](#)