Why Singletons are evil:

1. Singleton, as any global variable, creates hidden dependencies in your code. Non-static objects are passed to methods and thus the dependency is explicit. With singleton you have to inspect the code to see where it is used.
2. Violation of the “Single Responsibility Principle.” A class should not care whether it is a singleton; it should only be concerned with its business responsibilities.
3. Classes that use the singleton are tightly coupled with the singleton and are harder to test.
4. Singletons carry state with them as long as the program runs. This can hide bugs when testing as the state is carried from test to test.

<https://blogs.msdn.microsoft.com/scottdensmore/2004/05/25/why-singletons-are-evil/>