

## Code Review

Please review the following code snippet. Assume that all referenced assemblies have been properly included.

The code is used to log different messages throughout an application. We want the ability to be able to log to a text file, the console and/or the database. Messages can be marked as message, warning or error. We also want the ability to selectively be able to choose what gets logged, such as to be able to log only errors or only errors and warnings.

- 1) If you were to review the following code, what feedback would you give? Please be specific and indicate any errors that would occur as well as other best practices and code refactoring that should be done.
- 2) Rewrite the code based on the feedback you provided in question 1. Please include unit tests on your code.

```
using System;
using System.Linq;
using System.Text;

public class JobLogger
{
    private static bool _logToFile;
    private static bool _logToConsole;
    private static bool _logMessage;
    private static bool _logWarning;
    private static bool _logError;
    private static bool LogToDatabase;
    private bool _initialized;
    public JobLogger(bool logToFile, bool logToConsole, bool logToDatabase, bool
logMessage, bool logWarning, bool logError)
    {
        _logError = logError;
        _logMessage = logMessage;
        _logWarning = logWarning;
        LogToDatabase = logToDatabase;
        _logToFile = logToFile;
        _logToConsole = logToConsole;
    }

    public static void LogMessage(string message, bool message, bool warning, bool
error)
    {
        message.Trim();
        if (message == null || message.Length == 0)
        {
            return;
        }
    }
}
```

**Best Practice 001:** Remove unnecessary Using directives

**Best Practice 002:** It is unnecessary to set as "static" the private member variables, because its states won't be shared in other instances. Also the usage of Static variables has several disadvantages such as: Less Object Oriented, Object Lifetime is very long and Difficult to write Unit test

**Best Practice 003:** According official conventions, private member variables should use `_leadingUnderscore`. Only for public member variables is used PascalCase

**Best Practice 011:** Remove unused code

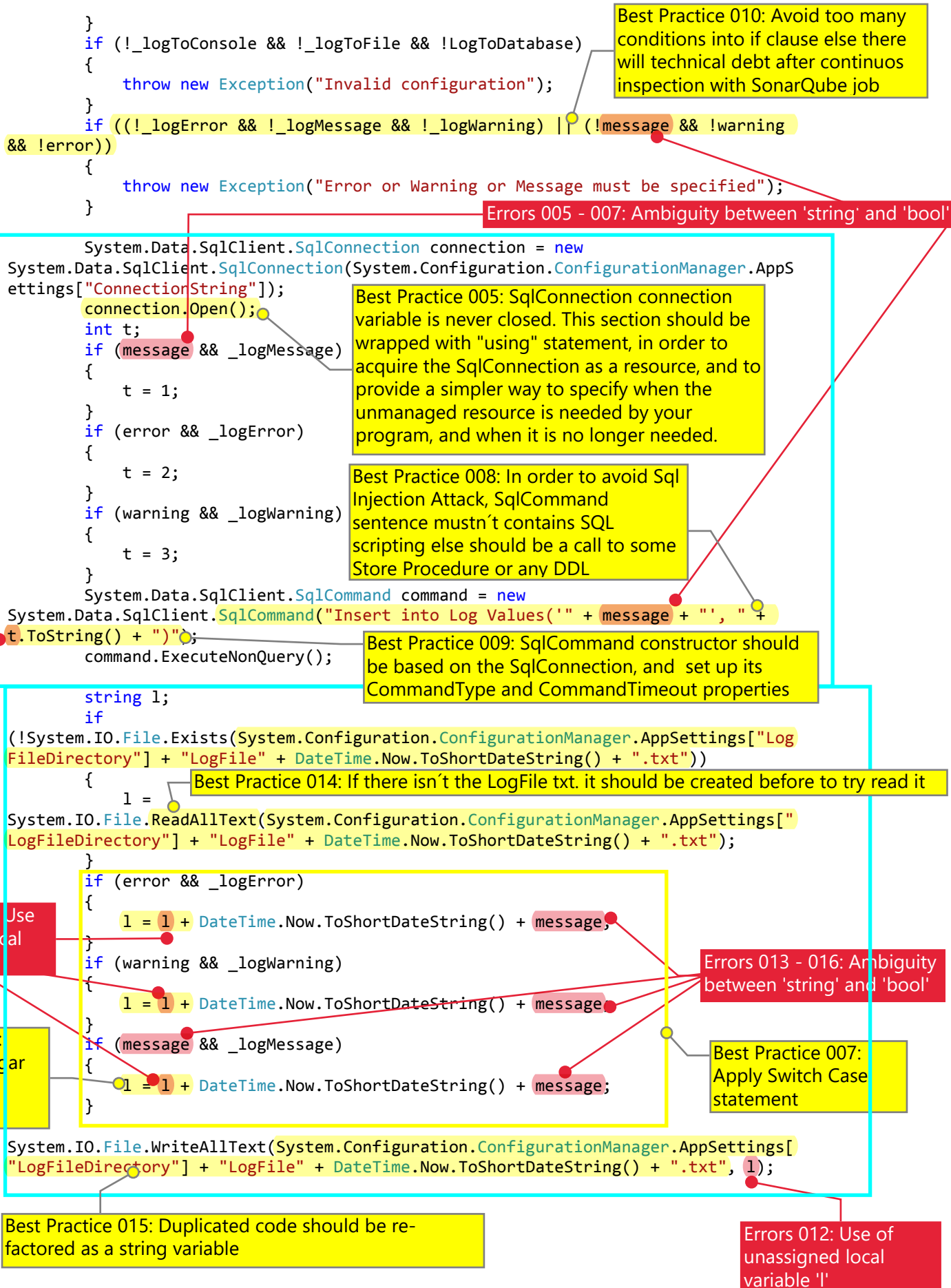
**Best Practice 012:** LogMessage method is coupled with properties of JobLogger class, whose get values through the constructor using Dependency Injection pattern. So, it won't be able to use it without to instance the class

**Best Practice 013:** the method should not be too complex, this's breaking the Single Responsibility Principle, but should be refactored to split in several class/methods applying Strategy pattern.

**Error 001:** The parameter name 'message' is a duplicate

**Errors 002 - 004:** Ambiguity between 'string' and 'bool'

**Best Practice 004:** In runtime, if message parameter value is null, the application will throw to NullReferenceException exception. So LogMessage method should manage NullReferenceException exception or focus its implementation with a value validation of this parameter before its first use



```
if (error && _logError)
{
    Console.ForegroundColor = ConsoleColor.Red;
}
if (warning && _logWarning)
{
    Console.ForegroundColor = ConsoleColor.Yellow;
}
if (message && _logMessage)
{
    Console.ForegroundColor = ConsoleColor.White;
}
Console.WriteLine(DateTime.Now.ToShortDateString() + message);
}
```

Best Practice 007: Apply Switch Case statement

Errors 017 - 018: Ambiguity between 'string' and 'bool'

For each skyblue box:

1. There isn't any conditional or switch structure which its respective section be wrapped according the context to log messages
2. It should be re-factored in other class/methods using Strategy pattern according its respective context to log