**Code Review**

**Gerardo Ramos – Aug 28/2018**

1. **Logical errors**
   1. On the LogMessage function, there is 2 variables with the same name “message” but with different data type (string and bool), this will cause an error.
   2. There is a string variable called “l”, but it wasn’t initialized at the declaration, so in the sentence l = l + DateTime…….. will present an error for use an unassigned variable.
   3. There is a int variable called “t”, but it wasn’t initialized at the declaration, so in the instruction to insert the log into the database if the variable has no value the instruction t.ToString() present an error for use an unassigned variable.
   4. Not assign the connection to the command, so when execute the command there will be an exception at the moment to run the instruction command.ExecuteNonQuery();
   5. Not close the connection.
2. **Best Practices** 
   1. Delete unused variable \_initialized
   2. Initialize variables “t” and “l”
   3. Use more specific name to variables “t” and “l”
   4. Comment de LogMessage function and their parameters
   5. Use string builder class to get better performance at the time to concatenate strings.
3. **Refactoring**
   1. Not put SQL sentence on the code (INSERT INTO… on this case), better use a stored procedure, or format the entire string with string builder to avoid SQL injection.
   2. Use try…. catch…. finally sentence to error handle any type of exception on the execution, for example a database time out or network connection.
   3. Use functions to separate specific task, save on database, save on file and show on console
4. **Custom Solution**
   1. It will be better use a pattern to simplify task, I recommend the factory pattern because if any moment we’ll need create a new log type (beside database and file) for example, call a web service, only we must create a new class inherited from the main abstract class.

