

# Logger Module Coding Test

## Log Task

You are requested to create a log component in 3 hours in any one programming language (Preferably C#, Python, or JAVA) , details of the requirement are below.

## Log Component

The component is a small module that can do asynchronous writing of strings to a file.

## Requirement

1: A call to **Write** on the interface must be as fast as possible so the calling application can get on with its work without waiting for the log to be written to a file.

2: If **we cross midnight a new file with a new timestamp must be made**

3: It must be possible to stop the component in two ways:

- Ask it to **stop right away** and if any outstanding logs they are not written
- Ask it to stop by the wait for it to finish writing outstanding logs if any

4: If an error occurs the calling application must not be put down. It is more important for the application to continue to run than lines not being written to log

## Unit test

The following must be proven by unit test:

1. A call to **Ilog** will end up in writing something
2. New files are created if midnight is crossed
3. The stop behavior is working as described above

## Suggestion on Projects organization

### **LogComponent** module (DLL or equivalent)

This is the project containing all the logic around logging things.

The ILog is the public interface

### **Application**

This project is the application using the LogComponent.

## Your task

Your task is to make the log component do what is specified in the Demands section and create the mentioned unit tests to prove it works.

## Things to do

- Make the design more SOLID
- Make the code more readable/maintainable
- find and fix bugs
- Create unit test

There are no limits in what you can do with the code. You can create new classes, new interfaces as long as the LogComponent will do as specified.

Regarding unit test: You are welcome to use any unit test framework of your own choice, or just do unit tests without a backing framework.

If you lack technical skills within any part of the task, you can write a textual description of what you would have done, and why.

## Things not to do

- Do not use or refer to existing/3rd party logging libraries. We want to see a unique and fresh approach.
- This library/module will be used in another library or application. Do not build REST API. It's not going to be used in a client-server architecture.

**Work outcome**

- code and unit test. If the task is too big to do within the available time please do as much as possible and describe the missing steps in text, pseudo code etc.
- A few words on the design

**How to share your work with us?**

- Simple! Please upload your work to a github and reply ALL to share the link to us.
- That's it. You are done!