

# Assignment Description

---

Create a program that reacts to user keyboard input by processing data (simulating processing by setting thread to sleep) and logs its actions in various ways. You can use CMake project and the file skeleton defined in this repository.

## Classes Requirements

---

### Logger

- logs the input string into a file "date\_log.log". Every logged line in this file should start with [**date-time**] prefix (current date and time in any chosen format)
- reads from file "date\_log.log" and output the number of lines in the file to STDOUT
- logs the input string into a file "functional\_log.log". Every logged line in this file should start with [**line-number**][**class-function-name**] prefix, where **class-function-name** represents the class-function name from which this log method was called
  - For instance if you log the very first time to "functional\_log.log" file from **Processor::loadFC()** method, the line in the log should start with [**1**][**Processor::loadFC()**]
- reads from file "functional\_log.log" and output every even line number to STDOUT
- logs the input string into STDOUT. Every logged line should start with [*time*][**class-function-name**] prefix only

### Processor

- has methods that simulate data processing by sleeping (use **std::this\_thread::sleep\_for()**) for a random float of time between 1.0 - 3.0 seconds
- logs using **Logger** into files or/and output to STDOUT every time processing method is called from the **Loader** class
- keeps track of its status - Processing Data or Ready To Process

### Loader

- reads user input from the keyboard and depending on content calls:
  - "fc" + ENTER - calls **Processor**'s process data method with logging this event into the "functional\_log.log" file
  - "out" + ENTER - calls **Processor**'s process data method with logging this event into STDOUT
  - "rdt" + ENTER - calls **Logger**'s read "date\_log.log" file
  - "rfc" + ENTER - calls **Logger**'s read "functional\_log.log" file
  - other input + ENTER - logs into the "date\_log.log" file file and STDOUT "Unknown command" message

- if **Processor**'s status is Processing Data while the process data method is called, "Processor class is busy" message must be logged to STDOUT instead
- every time it calls **Processor**'s process data method is called successfully it must measure the time it took to process the data (sleep duration) and output it into the "functional\_log.log" file and STDOUT

## General Requirements

---

- Each class (**Logger**, **Processor**, **Loader**) must be implemented in a separate header within the project
- All classes and its methods must be documented within the code (unless method's names are self-explanatory)
- Assume that the **Logger** class would be extended with different types of .log files and would process (read/write) those files in different manner
- The content of previously created log files must be overwritten when you restart your program
- Use atomic commits
- C++11 or above must be used
- prefer using only **std::** library (if other libraries are used explain why you made this choice); use **std::cin** for keyboard input
- CTRL+C terminates the program
- Write unit tests for the **Logger** class\*

*\* not obligatory to complete, but a plus for the interview discussion*