

Write a modern, cross-platform C++ library that provides the ability to log messages to different output channels via a logger. The logger should allow the application to add or remove output channels to and from the logger. Each type of output channel should also allow the application to specify which log levels that output channel should output. All messages written to the logger should be written to any output channel that has been added to the logger and configured to accept that message's log level.

The logger should support writing messages with the following log levels.

- 1 - Fatal
- 2 - Error
- 3 - Warning
- 4 - Info
- 5 - Debug
- 6 - Trace

Ordered by priority with 1 being the highest

Output channels should support setting a minimum log level, which will toggle on that log level and everything above it in priority. Alternatively, an output channel should allow the application to pick and choose which log levels to output.

Log messages should include a date/time stamp and a log level. Each output

Channel should allow the application to configure whether or not to include the timestamp and log level as a prefix to the message.

The library should provide the following types of output channels out of the box:

- Standard output: writes messages to standard output, ^{ONLY} log level info by default
- Standard error: writes messages to standard error, min log level Warning by default
- file: writes messages to a log file, min log level info by default
- log buffer: writes messages to memory, min log level info by default

The log buffer should allow the application to specify the number of messages to keep in the buffer and allow the application to retrieve messages and clear all messages in the buffer.

Nouns (Descriptor) [type]

- library [header]
- ✓ - message (log) [class]
- ✓ - channel (different, output) [base class]
- ✓ - logger [class]
- application [actor]
- type (output channel) [polymorphism]
- ✓ - level (log, minimum) [enum, parameter]
- ability [use case]
- ✓ - Extra [attribute]

- ✓ - Error [attribute]
- ✓ - Warning [attribute]
- ✓ - Info [attribute]
- ✓ - Debug [attribute]
- ✓ - Trace [attribute]
- priority [attribute]
- ✓ - timestamp [attribute]
- ✓ - prefix [return value]
- ✓ - box [N/A]
- ✓ - output (standard) [derived class]
- ✓ - error (standard) [derived class]
- default [constant]
- ✓ - file (log) [derived class]
- ✓ - buffer (log) [derived class]
- ✓ - memory [attribute]
- ✓ - number (messages) [attribute]

Verbs (subject, object) [type]

- provides (library, ability) [use case]
- ✓ - log (logger, message) [method] ←
- allow (logger, application) [use case] ←
- ✓ - add (logger, output channel) [method] ←

- ✓ - remove (logger, output channel) [method] ←
- allow (output channel, application) [use case]
- specify (application, log level) [use case]
- ✓ - specify (output channel, log level) [method] ←
- ✓ - output (output channel, message) [method]
- ✓ - write (output channel, message) [method] ←
- configure (application, log level) [use case]
- ✓ - accept (output channel, log level) [method], ... ←
- support (logger, writing) [use case]
- support (output^{channel}, setting) [use case]
- ✓ - set (output channel, minimum log level) [method] ←
- ✓ - toggle (output channel, log level) [attribute]
- allow (output channel, application) [use case]
- ✓ - pick/choose (output channel, log level) [method]
- ✓ - include (log messages, timestamp) [encapsulation]
- ✓ - include (log messages, log level) [encapsulation]
- allow (output channel, application) [use case]
- ✓ - configure (output channel, timestamp) [method]
- ✓ - configure (output channel, log level) [method]
- ✓ - provide (library, types) [polymorphism]
- allow (buffer, application) [use case]
- ✓ - specify (buffer, number of messages) [method]

- ✓ - keep (buffer, messages) [encapsulation]
- allow (buffer, application) [use case]
- ✓ - retrieve (buffer, message) [method]
- ✓ - clear (buffer, message) [method]