

# ReadStream Task

## Common requirements

1. Code must be written in standard C++ (98/11/14)
2. Use of OOP, RAII, exceptions and design patterns is encouraged
3. No other libraries except STL are allowed
4. Code should be supplied with correspondent unit-tests
5. Unit tests are allowed to use well-known unit-testing frameworks
6. Attention to code design has to be paid. Proper design is as much important as proper coding.

## IReadStream

Design an interface (abstract class) that allows to read data from the stream of any nature (file, network, audio etc.). Interface design should meet the following requirements:

- Non-blocking. If the data being requested is temporarily unavailable: the client of the class must not be blocked by the call.
- Streaming. The amount of data that the stream can provide is expected to be enormously large. Interface design must not induce neither the client nor its implementations to consume a lot of memory.
- Error handling must be exception-based.

## FileReadStream

Provide implementation for IReadStream that reads data from the file specified by path using your favorite OS API. Although the FileReadStream implements a non-blocking interface, the implementation itself doesn't have to be non-blocking.

In unit tests OS API usage is allowed.