

# Week 15 - I/O

## Topics covered in this week

- Streams vs. readers / writers
- character sets vs. encoding
- NIO
- Java 8+ features:
  - Files
  - streams of lines / characters
- Base 64
- encryption algorithms vs. hashing
- basic networking elements
- HTTP 2 Client

## Reading material

- [http://www.java2s.com/Tutorials/Java/Stream\\_Reader\\_Writer/Get\\_to\\_know\\_Java\\_input\\_output\\_stream\\_and\\_reader\\_writer.htm](http://www.java2s.com/Tutorials/Java/Stream_Reader_Writer/Get_to_know_Java_input_output_stream_and_reader_writer.htm) (inputstream/outputstream/reader/writer)
- <https://www.javacodegeeks.com/2018/07/java-nio-tutorial.html> (NIO)
- <https://www.baeldung.com/java-nio-2-file-api> (Files)
- <https://www.baeldung.com/java-char-encoding> (charsets, unicode, default charset)
- <https://www.baeldung.com/java-base64-encode-and-decode> (base64)
- <https://auth0.com/blog/how-secure-are-encryption-hashing-encoding-and-obfuscation/> (encryption vs hashing in general)
- <https://www.baeldung.com/java-cipher-input-output-stream> (encrypting/decrypting files)
- <https://www.edureka.co/blog/java-networking/> (networking in general AND in java)
- <https://www.baeldung.com/java-9-http-client> (Java test feature for new HTTP client)
- <https://www.dariawan.com/tutorials/java/multiple-requests-using-http11-vs-http2/> (HTTP1 vs HTTP2 client requests)

## Homework

Difficulty	Problem	Notes
HARD	Write a basic proxy server. Implement a simple HTTP client and server app, which communicate with each other via Sockets. The client will request a call to a webpage and the server should return the content of the requested webpage.	See <a href="https://www.edureka.co/blog/socket-programming-in-java/">https://www.edureka.co/blog/socket-programming-in-java/</a>
EASY	Write in a file 4 integers and then reread the third and fourth ones using random access to the file.	
EASY	Read an arbitrary file, compress it to another file and display the <i>compression ratio</i> .	
MEDIUM	<p>Write a small application that reads a <i>Latin1</i> encoded text file and writes its contents into a <i>UTF-8</i> one (transcodes it).</p> <p>Now try to do the reverse operation as well: read a <i>UTF-8</i> encoded file text (not the one generated before) and try to transcode it to <i>Latin1</i>. Why is this not possible? Output to the console or to another file all the characters that can not be transcoded.</p>	