**Assignment 1 - Simple DB with Hashmap-based Index**

* Build a simple\_db in the programming language of your choice.
  + Implement a Hashmap-based index.
  + Implement functionality to store your data on disk in a binary file.
  + Implement functionality to read your data from disk again, so that you can reinstantiate your database after a shutdown.
* *Hint:* You do not want to serialize and deserialize the an in memory Hashmap containing all data directly. Instead, you in memory index based on a Hashmap contains information on where in you database file a piece of information is stored.

**Hand-in procedure**

* Provide all code and documentation for this assignment in a repository on GitHub.
* Create a Markdown (.md) file called README.md in the root of your project.
* That README.md describes what this project does and how to make it work. That is, you reviewer has to be able to clone your project, build it -you have to define steps for how to do that and what dependencies are required-, and use it.
* Hand-in a link to your repository on www.peergrade.io.
* Hand-in at latest on Sunday 3. Feb. 16:00.

**Review procedure**

* Log onto www.peergrade.io with your school email address.
* Finish your review on www.peergrade.io at latest on Tuesday Feb. 5th 23:00.
* Make use of the review criteria below when giving feedback.
* See the message that you received on your school email to know whom to review.

### Review Criteria

Provide feedback on each of the following bullet points.

* Does the provided solution to the assignment work as described in the README.md?
  + If not, describe briefly what information is missing or what is wrong in the program.
* Describe briefly, how the provided solution compares to the description of Hashmap-based indexes on page 4 of the PDF in <https://github.com/datsoftlyngby/soft2018spring-databases-teaching-material/blob/master/literature/session_1.zip>.
* Does the provided solution allow for storage and reinstantiation of the Hashmap-based index when the database is shutdown and started respectively?
  + If not, provide information on what is missing and how that could be implemented.