MEMORIAL UNIVERSITY
Faculty of Engineering and Applied Science

ENGI 9818 Computer Software Foundations

Fall 2023

Assignment 1

Warning

Zero-tolerance on academic dishonesty. Academic dishonesty will, as a minimum, result in a mark of zero for the offending work. Academic dishonesty will be reported to the Associate Dean. Academic dishonesty includes copying from another student's work or allowing another student to copy from one's own work, consulting with any unauthorised person during an examination or test, using unauthorised aids (including AI, e.g., GPT-derivatives), and presenting the ideas or works of another as one's own. If you have any doubt about whether something constitutes academic honesty, consult with the instructor.

Question

You are tasked with creating a simple Database Management System (DBMS) for managing student records at a university.

Requirements

- Implement a data structure of your choice (either B-Tree or Skip List) to manage student records and the indices.
- Create a hash function to build an index based on the overall score (midterm + final).
- Develop another hash function to build an index based on the student number.
- An example DBMS class is provided which facilitates querying using your chosen data structure.
- A Student class is provided below to represent individual student records.
- Your implementation shall have good performance.
- You need to write at least 3 different test cases. Do not test with large amount of data.

Submission

It is extremely important that you are following the naming conventions for your class, tests, files, and submissions. If the automatic system fails to recognise your submissions or failed to execute them, your assignment will be treated as NOT SUBMITTED.

Late submissions are NOT allowed and will NOT be accepted.

Requirements

1. Your submission must be a zip file and the file name must be your student number. Example: 202312345.zip
2. Your submitted zip file must contain “DBMS.java” and “DBMSTest<student number>.java”. The structure shall like:
   ```
   202312345.zip
       |- DBMS.java
       |- DBMSTest000000000.java
   ```
3. The “DMBS.java” file must include a public class, “DMBS”, which implements the “DMBSInterface” interface.
4. You can change anything in your submission.
5. No third-party libraries or dependencies.
6. Your code must be working with JDK 17 and up.
7. The “App.java” file include a basic example. Your program shall be, at minimal, working with the example.
8. All your own tests shall pass.
9. Code style is important. You need to follow Java naming conventions and ensure there are no unnecessary extra new lines, spaces (e.g., additional spaces at the end of lines, a line with only spaces), etc.
10. Do not declare packages. You must use the default package.

Hints (you don’t have to read or follow this section)

1. You can either implement the storage backend with B-Tree or Skip List.
2. Making a hash function work is easy, but designing a good hash function is difficult.
3. A possible way toward the solution:
   a. First, identify the fields for indexing, e.g., Student Number, Score.
   b. Design hash function for each type of fields
   c. When inserting into the storage backend, first calculate the hash value for the index (e.g., Student number, Score), then insert both index and the Student object as Pair into the location specified by the hash value
4. When in doubt, debug your code line by line to understand how things work.