# **CSCI X370: Database Management**

## **Project 3: Index Structures - Linear Hashing**

Use indices TreeMap (in Java), HashMap (in Java) and LinHashMap (your own implementation) to speed up the processing of Select and Join in your earlier project implementation. Indices must be integrated and used in the Table class.

**Implement LinHashMap**:

It has already been partially implemented in LinHashMap.java (at eLC) and what you need to do is to fully implement this index structure. The main method of LinHashMap.java is used for testing; however, since functions are partially implemented, it will not give the proper output. So, you should be able to see the desired output after the completion of implementation. Download the starter source code from the eLC and compile it. Upon making sure that it is working properly, you can start implementing. After implementation, you can compile and run your code to verify if the outputs are correct.

**Performance Evaluation:**

Compare the speed of (a) NoIndex, (b) TreeMap, (c) HashMap and (d) LinHashMap. Plot the speed/performance in terms of response time in ms vs. number of tuples. The project will require the use of TupleGenerator.java, TupleGeneratorImpl.java and TestTupleGenerator.java.

Compare these four for (i) Select Operations: "σ id = v (Student)" and (ii) Join Operations: "Student join id = studId Transcript".

**LinHashMap.java:** This file contains the class which implements hash map that use the Linear Hashing algorithm. Basic methods have been implemented and a test function is provided for testing. Implement the code in this file where you find “// TO BE IMPLEMENTED”.

**Table.java**: This file contains Table class which implements relational database tables. Five basic relational algebra operators are provided: project, select, union, minus and join as functions. You need to implement indexed-select and indexed-equi-join for performance evaluation. For indexed-select, it is simple (no more than 5 lines of code) and non-indexed version is provided for evaluation (see non-index.java). Non-indexed version of equi-join is also provided to make the implementation quick and easy (see non-index.java).

**What to submit:** Please submit

* all source code
* A half-page summary written by the team leader outlining the overall status of the project which will include an assessment of the contributions of the group members.
* Performance evaluation: Graphs and relevant data points.

The readme file should contain: your names, how to compile and run your code and other specifications you want to make (including your database access information for testing purposes). Please pack all your files in a zip package with the file name: project3\_ last names of your group members. For example: project3\_Miller\_Arpinar.zip

**How to Submit:**

Submit your ".zip" file using ELC. Only team leaders need to make a submission.

Do not place your solution on a public web site. Submit your own work and follow the course misconduct policy.