

CSCI 230 – Homework 7

Collaboration policy: **Individual Assignment**

Total Points: **170**

Source Code

The provided Java classes are:

- `OpenHashing.java`
- `ClosedHashing.java`
- `HashTable.java`
- `HashTableKeyException.java`
- `HashSimulator.java`

Provided text file

- `words.txt`

Under no circumstances are you allowed to create a new `HashSimulator`, `OpenHashing`, `ClosedHashing`, `HashTable`, or `HashTableKeyException` class. Lastly, you may not modify the `words.txt` file.

The `List` interface and `ArrayList` class developed in a previous homework assignment will be reused. Under no circumstances are you allowed to modify these two files, you must use them **as is**. You **may only** modify the `TODO` sections in the `HashSimulator`, `OpenHashing`, `ClosedHashing`, and `HashTable` classes. In particular, you **may only** modify the methods listed in Part 1, and under no circumstances are you allowed to remove, add, or modify any other line of code in this class **this include instance variables, class variables, constants, etc.**

Lastly, you **may not** change the package structure! Specifically, `edu.cofc.csci230` cannot be removed or modified.

Part 1

In the `HashTable` class please fully implement the methods listed below:

- `public int calcHash(String key)`

In the `OpenHashing` and `ClosedHashing` classes please fully implement the methods listed below:

- `public int search(String key) throws HashTableKeyException`
- `public void insert(String key) throws HashTableKeyException`
- `public int delete (String key) throws HashTableKeyException`
- `public double loadFactor()`
- `public double successfulSearches()`
- `public double unsuccessfulSearches()`

In each method listed above, you will see a `TODO` comment, this is where your coding solution is added. In the provided source code, numerous comments are given; please ensure you read them carefully. Additionally, in the supplemental course textbook (PDF provided in content section on OAKS) provides

explicit details of each hashing data structure along with the metrics (load factor, successful searches, and unsuccessful searches) that are used to assess the time efficiency.

Part 2

In the `HashSimulator` class there is a main method. In the main please complete the eight coding steps (see TODO comment) for **both** hash data structures and **both** hash functions.

Submission

Create one zip file that **only** includes the completed `HashSimulator`, `OpenHashing`, `ClosedHashing` and `HashTable` files. If you have any questions about the submission policy, you must resolve before the due date. Lastly, please plan appropriately, asking questions the day the assignment is due (within 12 hours) is too late. Please try to resolve any questions at least 2 days before the due date.

The name of the zip file must be your last name. For example, *ritchie.zip* would be correct if the original co-developer of UNIX (Dennis Ritchie) submitted the assignment. Only assignments submitted in the correct format will be accepted (no exceptions).

Please submit the zip file (via OAKS) to the Dropbox setup for this assignment by the due date. You may resubmit the zip file as many times as you like, Dropbox will only keep the newest submission. Per the syllabus, late assignments will not be accepted – no exceptions. Please do not email Luis or me your assignment after the due date, we will not accept it.

***** Lastly, this is an individual assignment, i.e. no collaboration is permitted. Plagiarism will not be tolerated. Submitted solutions that are very similar (determined by the instructor) will be given a grade of zero. Please do your own work, and everything will be OK.**

Grading Rubric

Solution compiles and runs	10 points
Eight steps in main method (5 points each) * 4	160 points

In particular, each data structure will be graded as follows.

- Does not compile and or run: 0 of 170 points
- Compiles and runs: 10 of 170 points
- All steps in main method produce correct answer: 170 of 170 points