

LAB 5 (HASH TABLE)

CSC 172 (Data Structures and Algorithms)
Fall 2022

University of Rochester

**Due Date: Monday, October 24th 2023 @ 8:00
am**

Introduction

The labs in CSC172 will follow a pair programming paradigm. Every student is encouraged (but not strictly required) to have a lab partner. Every student must hand in their own work but also list the name of their lab partner if any on all labs.

In this lab, you will work on the Hash Table/Map ADT. Build a Linear Probing Hash Table. Start with the `URHashTable` class code below. Make a concrete class and include the helper methods that are commented out.

```
abstract public class UR_Table<Key,Value> {

    private static final int INIT_CAPACITY = 5 ;

    protected int n; // size of the data set
    protected int m ; // size of the hash table
    protected Key[] keys;
    Value[] vals;
    int inserts, collisions;

    // Constructors
    //     public UR_Table() {}
    //     public UR_Table(int cap) { }

    abstract public void put (Key key, Value val) ;

    abstract public Value get (Key key) ;

    abstract public void delete(Key key) ;

    abstract public int size() ;

    abstract public boolean isEmpty() ;

    abstract public boolean contains(Key key);

    abstract public Iterable<Key> keys() ;

    // Useful helpers
    // private int hash(Key key) ;
    // private void resize(int capacity) ;

}
```

Before you start

Note:

- Be sure to include a Unit Test program that demonstrates how your program works.
- You MUST implement `Iterable<T>` interface for the Key set in the class you implement.
- Please write any assumption made. For example: describe what the return value means.
- Your methods must handle all the corner cases gracefully — for example, throwing exceptions with detailed explanations or returning values indicating the error in case the operation is not permitted. The comments should clearly state the issues and the remedies involved. In short, no illegal operation should be permitted and the list and all its parameters should be in a valid state.

Submission

Submit a single zip file `Lab5.zip` containing the class you implement, a unit test file containing a main method, a README file, and your class files. Your README should describe how to run your unit tests. Upload this file at the appropriate location on the Blackboard system at `learn.rochester.edu`.

1. All source code files should contain author and partner identification in the comments at the top of the file.
2. All source code should include links to any internet resources you used outside of course materials.
3. A plain text file named README that includes your contact information, your partner's name, a brief explanation of the lab (a one-paragraph synopsis. Include information identifying what class and lab number your files represent.), and one sentence explaining the contents of any other files you hand in. You can also share testcases those you ran.

Grading

Total: 100 pts

- 10 pts for README file, proper commenting, and error handling.
- $(11 * 7 =) 77$ pts for 12 methods shown in the sample code.
- 13 pts for your main method containing your unit testing class (main method, tests, etc)