ECE250 - Project 2
Hashing
Design Document

Leah Burgess, UW UserID:l2burges
November 8th, 2022

Overview of Classes

Class: Student
Description: Represents each individual student as an object in the dynamically allocated array. Each student is defined by their student number and last name.
Member Variables (all private): string name (their last name), unsigned int x (their student number)
Member Functions (public):

- Setters for the string name, and unsigned int x
- Getters for the string name, and unsigned int x

Class: doubleHash
Description: Represents the students as a dynamically allocated array and contains the methods for managing the hash table (insert student, search student, delete student).
Member Variables (private):

- Int size: the max size of the array
- Int currentSize: the current size of the array (the number of students in the hashtable)
- Student *student: dynamic array as a pointer filled with the student objects that acts as a hash table

Member Functions (public):

- String insert(unsigned int x, string name) - If the number of students in the hash table are less than the max size, check if the space from the first hash function is empty. If it is empty, insert and return success. Otherwise, check if the student already filling the spot is the same student number. If it is, return failure. Otherwise, calculate (h1 + i*h2)modm and repeat the process from that spot in the hash table. If the table is full, return failure.
- String search(unsigned int x) - While the number of iterations is less than the size of the table, if the student number from the first hash function is equal to x, return “found last name in position”. Otherwise, calculate (h1 + i*h2)modm and repeat the process from that spot in the hash table. If you haven’t found the student number and the while loop ends, return failure.
- String deleteHash(unsigned int x) - While the number of iterations is less than the size of the table, if the student number from the first hash function is equal to x, set the name of the student to an empty string and the student number to 0 and return success. Otherwise, calculate (h1 + i*h2)modm and repeat the process from that spot in the hash table. If you haven’t found the student number and the while loop ends, return failure.

Class: chainingHash
Description: Represents the students as a dynamically allocated array of vectors and contains the methods for managing the hash table (insert student, search student, delete student, print chain).
Member Variables (private):

- Int size: the max size of the array
- vector<Student> *student: dynamic array as a pointer filled with vectors that hold the student objects that acts as a hash table

Member Functions (public):

- String insert(unsigned int x, string name) - Check if the vector’s size in the position of the array from the first hash function is 0. If it is 0, insert and return success. Otherwise, check if