Data Structures (2028C) -- Spring 2025 – Lab 10

Topics covered: Hashing

Lab due: ***Sunday, April 13 at 11:55PM for Monday Section***
Tuesday, Apr 15 at 11:55 PM for Wednesday Section

Objective:

The objective of this Lab is to explore creating and using Hash Tables.

Task 1: Create a Hash Table that uses linear probing.

1. Create a new project. You can name this whatever you like.
2. Design and implement Hash Table class using linear probing as described in class. The data being stored is a Student class (you will need to create this). The data being stored in the array in the Hash Table is pointers to a Student.
    a. The class should have the following methods fully implemented.
        i. Constructor – This should have an overload indicating the maximum number of items the Hash table can store. The default value is 100.
        ii. Hash – This should be a private function that accepts a string and returns an int. For this, we are going to add the sum of all the ASCII values for the passed in string and modulus by the maximum size of the table.
        iii. AddItem – adds an item from the list.
        iv. RemoveItem – searches the list for the given item. If found, it removes it from the list and returns it. If not found, it returns a null pointer.
        v. GetItem – searches the list for a given item. If found, it returns a pointer to the item but doesn't remove it from the list.
        vi. GetLength – returns an int indicating the number of items in the table.
        vii. Destructor
    b. All items passed to or from the class should be done so via a pointer rather than by value.
    c. Make sure you don't have any memory leaks.

Complete this before moving on to task 2.

Task 2: Create a student class. It needs a First Name, Last name, MNumber, and Major. You will need to add the overload of the string object conversion. This object conversion should return the MNumber.
Complete this before moving on to Task 3.

Task 3: Modify your test program from Lab 7, Task 3 (test program) to test your Hash Table class. Include a screen shot of some of this testing in your lab report.
Complete this before moving on to Task 4.

Task 4: Create a derived class from your Hash table that implements a chained hash table. Use your linked list class from Lab 7, Task 1. Modify your test program from Task 3 to test this new class. Include a screen shot of some of this testing in your lab report.

Complete this before moving on to task 5.

Task 5: Measure the performance of the linear probing and chained linking implementations of a hash table. To do this, modify both classes to keep track of the number of times an item is compared in the chain for chained linking[i] and the number of times an additional index is checked for linear probing (you only need to do this for the GetItem but you can do it for both if that is easier). Create a test program that adds 50 randomly generated students (follow appropriate MNumber rules) to both hash tables then calls GetItem on the same student. Record the total number of times the Hash Table needed to complete the 50 operations. Repeat this measurement with an array size of 150, 200 and 250. Include a table of the results from the 4 trials for the 2 different classes in your lab report. Include discussion about your findings including recommendations on table size for each hash table size.

Lab Submission:

1. Write a lab report including the following information:
    a. A description of the objectives/concepts explored in this assignment including why you think they are important to this course and a career in CS and/or Engineering.
    b. The sections from each task indicated to be included in the lab report.
2. Include all source code from all tasks, input and output files (if any), and any special instructions to compile and run those programs.
3. Include a discussion of what modifications you needed to make for Task 2 and Task 3.
4. Include a discussion of why you think the results for Task 5 were what they were. This should suggest ideas for further investigation.
5. Package all files in a single zip folder and submit the file to Canvas.

Lab Grading:

1. 20% - Lab attendance
2. 20% - Task 1 has been correctly implemented and meets all requirements.
3. 5% - Task 2 has been correctly implemented and meets all requirements.
4. 5% - Task 3 has been correctly implemented and meets all requirements.
5. 15% - Task 4 has been correctly implemented and meets all requirements.
6. 20% - Task 5 has been correctly implemented and meets all requirements. The discussion makes sense and includes suggestions for further investigation.
7. 15% - Lab report contains all required information and is well written.

If program fails to compile, 0% will be given for that Task.

[i] You may need to modify the Linked List class to achieve this.