# California State University, Fresno

# DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | **Algorithms & Data Structures** | | | Semester: | **Fall 2021** |
|  | | | | | |
| Points |  | Document author: | **Saishnu Ramesh Kumar** | | |
|  | Author’s email: | **saishnu\_rk@mail.fresnostate.edu** | | |
| Laboratory number: | **Section 1, 11am to 12:50pm** | | |
|  | | | | | |

1. **Statement of Objectives**

This lab assignment was to do with reading the file, RomeoAndJuliet.txt into a C++ program to detect how many unique words are in the story as well as getting the ratio, cost[i] / updates[i]. This program uses the open addressing format to prevent collisions from occurring but in the linear probing format whereby the program would check each index to see if it’s empty, if the index is not empty, it would move on to the next and so forth until it finds an empty index to place the data into.

**2. Experimental Procedure**

A hashArray class was created where all the functions were declared and were called outside of the class. The macro, HASHSIZE was created to store the total number of unique words and since the total number of unique words were 3,684 we were told to use 4001. Inside the hashArray class, the private data members mainly stored the key, hashTable[HASHSIZE], update[HASHSIZE], and cost[HASHSIZE]. The public data members mainly contained the constructor, destructor as well as the functions. The hash function retuned the index for the key, the updates function helps with inserting the key in the index and to update the list accordingly. The size function deals with returning the number of unique words in the text file. The ratio function gets the ratio of cost[a] / update[a] and prints out the ratios in the output terminal. The main function contains the file call where the text file is called into the program to be used and also helps output the total number of unique words and the ratio to the output terminal.

**3. Analysis**

It can be seen that the number of unique words in the file are 3,684 and the ratios are all 1.00 (set to two decimal places) that are separated by commas.

Terminal Screenshot Output:

Table

Description automatically generated with medium confidence

**4. Encountered Problems**

I encountered some compiler errors and a number of times I faced some dump stack errors as well but was able to fix the issues after going back through the code as well as checking some websites online.

**5. Conclusions**

From this lab I have learnt and got a better understanding about Hash Tables and how they are intended to work. I have also learned how to import a text file to be read inside a C++ file and how the words in the text file are basically scanned through using the program to see how many unique words there are in the file.

**6. References**

Slides provided by TA.

<https://www.softwaretestinghelp.com/hash-table-cpp-programs/>

<https://www.educative.io/edpresso/how-to-implement-a-hash-table-in-cpp>

<https://www.geeksforgeeks.org/c-program-hashing-chaining/>

<https://cplusplus.com/forum/general/60901/>

<https://www.geeksforgeeks.org/hashing-set-3-open-addressing/>