CSCE 221 Assignment 5 Cover Page

First Name Chris Last Name Comeaux UIN 622006681

User Name cmc236 E-mail address cmc236@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more on Aggie Honor System Office website: `http://aggiehonor.tamu.edu/`

Type of sources		
People	Peer Teacher	Dr. Teresa Leyk
Web pages (provide URL)	http://www.cplusplus.com/	http://stackoverflow.com/
Printed	Programming Principles and Practices using C++ by Bjarne Stroustrup	
Other Sources	http://regexr.com/	

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.
On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.

Your Name Chris Comeaux Date 4/4/2016

Assignment 5 Description

In programming assignment 5 the students had to create a program that could read in a csv file, parse the data using regex, hash the data into a hash table, and then finally write the updated data to another csv file. In this case, the input file was a file containing the name, email, UIN, and a quiz grade of a student. The first part of the program extracts the UIN and grade from input.csv and hashes the data by using the UIN as a hash value and the grade as a value. Next, the program reads in another csv file (roster.csv) and extracts the name, email, and UIN from the file. It then searches the hash table using the UIN to see if a grade exists for that student. If a grade exists, the program will write the name, email, UIN, and grade to an output.csv. If the grade does not exist, the program will only output the name, email, and UIN. The output file should look just like roster.csv only with the grades added to the students who took the quiz.

Data Structures and Algorithms

HashTable: This is a user defined structure that was used to describe a hash table; it has one member function and one helper function. HashTable is simply just a STL vector of singly linked list and uses the chaining method to deal with collisions. It also contains a reside funtion that calls vector reside. This member function was not necessary, however it cleans up the implementation in the main file. It also has a helper function that is just the hash function defined as UIN%size.

SinglyLinkedList: SinglyLinkedList is a very simple class that describes a singly linked list only for the use for the chaining method. It is made up of a sequence of SListNodes. It is made up of 2 SListNode pointers, head and tail, that points to the first and last node respectfully. It has few member functions to insert a new node at the end of the sequence, to remove all the nodes(used in destructor), to search for a key return the key's value, and a function to check if the list is empty. It is a friend class of SListNode so it is also able to access its private members.

SListNode: SListNode is a class that describes a node in a SinglyLinkedList. It has a key, a value, and a pointer to the next node. It has one member function that returns the next node.

Description of Input and Output Data

The input are 2 csv files that have 3-4 fields in each which are name, email, UIN, and grade (for the input.csv only). The output is also a csv file with the same 4 fields, however the grade field in not filled in for every line. A few assumptions were make about the input data. For example, an assumption was made that the names did not include any middle names and were of the format (Name Name). Another one is that is no type of error in the input files (i.e. wrong UIN or grade format). The program does not receive any input from the user so no assumptions were imposed on the program, however, the program does assume that 2 input files named, input.csv and roster.csv, are in the directory with the program.

Testing

To test for correctness, I first started by checking all of my regular expressions. I would search each individually and output the matches to the console to make sure they were matching the correct fields. Once this was done, I completed writing the rest of the program. Once I got program to terminate and produce the desired output, I started to study the hash table to make sure it was working correctly. First, I checked to see if the vector was re-sized to the correct size by outputting the size of variable at ever step in the loop that calculated the input file size. Next, I made sure that the hash function was distributing the data as evenly as possible. To do this I inserted a size accumulator into the SinglyLinkedList class and every time a UIN was inserted I updated the size. I then looped through the whole vector and output each list size to the console (2 was the largest size). Finally, I compared both input files to the output file to make sure no data was changed in any way.

C++ Features and Standard Library Classes

I did not use any generic programming features in my program, however I did use some object oriented features. For example, I created an object to represent a hash table, a linked list, and linked list nodes. Also, I used abstraction to hide private members and hide the implementation of linked list behind the hash table class. I used 2 standard library classes. I used the STL vector to create the hash table. Also, I use the STL Regex class to create my regular expressions and extract the fields from the csv files.

Running Time for Hashing Algorithms

Hash Function: The running time function for the hash function would be f(n) = 1 which is O(1).
Search: The running time function to search would be f(n) = n+1 which is O(n). However, since the linked lists are so small (max size = 2) they actually run in constant time so O(1).
Insert: The running time function to insert data into the has table would be f(n) = 2 which is O(1).
Destructor: The running time for the desctructor (remove_all in a linked list) would be f(n) = 2n+2 which is O(n).
Resize: The running time for the reside funtion would be f(n) = 1 which is O(1).

Conclusion

The purpose of this assignment was to teach students about hash tables and the chaining method for dealing with collisions. Also, it thought students about csv files and how to use regex to parse csv's. This assignment additionally built on students knowledge of linked lists and expanded their understanding of the implementations of linked lists. This assignment will help students with future projects using hash tables.