CSCE 221 Assignment 5 Cover Page

First Name: Matthew Last Name: Stevens UIN: 924000693

User Name: Mattizrawr E-mail address: Mattizrawr@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/

Types of Sources				
People	Joshua Langley	Nicholas Wong	Other lab members	
Web Pages	www.piazza.com	https://docs.python.org/2/library/re.html		
Printed Material	N/A			
Other Sources	N/A			

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: Matthew Stevens Date: 4/5/2016

– Assignment number and its description.

Assignment: CSCE 221 Assignment 5

Description: The purpose of this project is to write a program to help record every students' grades in a spreadsheet roster. When TAs collect results from an online quiz platform, the platform can only provide the grades of students who have submitted their quizzes. If a student skips them, the online platform will not keep track of their grades. Therefore, when a TA downloads data from the platform, he/she will only have the submitted results, which is actually a spreadsheet.

– Description of data structures and algorithms used by your program.

A hash table uses an array of linked lists as the data structure containing the information that we access.

HashMap(): makes an index for the linked lists which are put into the array

int HashFunc(int key): this is the key when organizing the uin's, it just uses mod table_size, which in this case is 100. This gives each uin a specific and unique location.

void Insert(): this is used for inserting new nodes in the hash table (array of linked lists) it does this by reassigning the previous and next pointers for the nodes before and after the insertion point.

void Search(): searches through the file using a key to compare it to an index which in turn returns the value which you are looking for.

– Description of input and output data. List all restrictions and assumptions that you have imposed on your input data and program.

For the input, in lab, we were told that the first two columns could be anything, the third column has to be 9-digits, and the fourth column has to be a number between 0 and 99. The output outputs a file, which combines the roster file with the input file and updates the grades of the students. The restrictions we placed were that the grades had to be 0-99 and the uin had to be 9-digits.

– How have you tested your program for corrections?

I did a few things. First, I placed a ton of couts that would cout a number, in order to find out where my code ends. Second, I ran a debugger and back traced it in order to find out what went wrong during a seg fault. Lastly, in order to test my collisions in the hash table, I added a few numbers to see if the collisions would do anything, and they worked.

– Which C++ features or standard library classes have you used in your program?

<iostream> <cstdlib> <string> <cstdio> <fstream> <regex>

Classes, object oriented features, and function calls using parameters.

– What is the expected running time of the hashing algorithms?

Expected running time is O(n), worst case is $O(n^2)$

– Write your conclusion.

In conclusion, this assignment was to deepen our understanding of hash tables, as well as, grant me experience for implementing said hash tables. This assignment also allowed me to learn regex, which I have never been taught before. I also learned how to read in from different file types. The output file was a spreadsheet that updated the students' grades after receiving the input of quiz grades for the students. This assignment also taught us the exact purpose of hash tables, which is to take huge numbers and compare then by using the furthermost 3-digits and colliding in order to sort them effectively.