# CSCE 221 Assignment 5 Cover Page

First Name: Colin Last Name: Legge UIN: 223006618
User Name: calegge714 E-mail: calegge714@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 Source | | | | |
| People | | | | |
| Web Pages | http://stackoverflow.com/questions/16446665/c-read-from-csv-file (General info on reading data in from a .csv) | | | |
| Printed Material | Programming: Principles and Practice Using C++, 2nd Ed. by Bjarne Stroustrup (regex reference) | | | |
| Other Sources | | | | |

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: Colin Legge Date: 5 April 2016

# Report

In this programming assignment 5, I was tasked with writing code that would utilize a hash table to store input values from a .csv file. Data was to be read from ‘input.csv’ and parsed for the UIN and grades and then the grades were to be stored in a hash table with the corresponding UIN as the key. Then ‘roster.csv’ would be scanned and parsed for the UIN and if it matched a key in the hash table, the corresponding grade would be inserted into a new ‘results.csv’ along with the rest of the student info. Otherwise, only the student info would be copied. In order to run my program, navigate to the directory that holds all of the files submitted using the TAMU UNIX server, and type ‘make’. My makefile should compile all necessary files for you. Now, type ‘./main’ and my code should run. It outputs the progress of my code along the way so that you may see what values are being stored in the hash table and under what keys.

The main data structure used in this assignment was a hash table. The hash table essentially maps keys to different values, so that there is a unique key for each value. The key of the hash table to created by using the modulus operator to find the remainder when dividing the desired key with the table size. Linear probing was used if there was any overlap in key generation. In this assignment, a table size of 17 was used since it is the smallest prime number larger than the number of inputs into the hash table, which lowers the probability of chaining being necessary. Also used in this assignment were vectors. Vectors were used as a way to store the various keys and grades for error checking during testing. The lines of ‘roster.csv’ were also stored in a vector to optimize the copying of the contents from ‘roster.csv’ to ‘results.csv’. The regex_search algorithm was used, which scours an input for a pre-defined pattern and stores any instances of that pattern that it encounters.

The input data of ‘input.csv’ consisted of 4 columns containing the student’s name, their email address, their UIN, and their grade on the assignment. ‘roster.csv’ contains all of these as well, except for a column for grades. When opening these files with Notepad++, each row is listed on its own line with each column separated by a comma except for the last data entry in ‘input.csv’, which is followed by blank space. I used this to my advantage by using the getline() function to read in a line of the input files and then used regex to parse each individual line for the data values that I was looking for. The output data in ‘results.csv’ is the same as ‘roster.csv’, except it also contains the grades from ‘input.csv’ in the same rows as the corresponding UIN’s as stored by the hash table.

During testing, I faced some problems. Most of my issues came from regex, since I was not as familiar with it as other C++ functionalities. After much testing or both the syntax and logic of my code, my code functions as desired as long as the input data is in the same format as the input files provided to us for this assignment.

Used in this assignment were the C++ features of regex in the form of the regex_search() function, as well as the erase() function to remove characters from read strings and stoi() and to_string() functions to change the data type of some values. An STL class used was the vector class in order to store the keys used for the hash table and the lines from ‘roster.csv’ to simplify the copying from ‘roster.csv’ into ‘results.csv’.

The expected running time of the Insert() function of my hash table class is O(1) and the expected running time of the Search() function is also O(1) since the hash table is a predefined size and no resizing of my hash table is being done.

After completing this assignment, I realized just how useful hash tables can be in the organization of data structures. I also learned how regex can be used to parse information from a read file, which will prove to be very useful to me in the future. After utilizing this data structure and I/O method, I feel that I was able to correctly and sufficiently meet the guidelines of the programming assignment with minimal errors, if any.