The Programming Assignment Report Instructions
CSCE 221- PA5

1. The description of an assignment problem.

   The purpose of this programming assignment is to create a program to help record every students' grades in a spreadsheet roster. The online quiz platform can only export data about students who have taken the quiz. Students who decided to skip or not do the quiz are not listed. This program will output a sheet that incorporates the grade into the roster.

2. The description of data structures and algorithms used to solve the problem.

   (a) Provide definitions of data structures by using Abstract Data Types (ADTs)

      i. **HashTable** - a data structure is an array of linked lists with mapped keys to values.

   (b) Write about the ADTs implementation in C++.

      The hash table was used to store the students UIN and associated quiz score. The UIN was hashed to create an index location. This location was then mapped to the associated quiz score. If there were collisions, the chaining method would be implemented.

   (c) Describe algorithms used to solve the problem.

      i. **hash.hash()** - this function was simply UIN modulo TableSize
      ii. **hash.insert()** - this function first determined the index position of the UIN by calling **hash()**. Next, if the item at HashTable[index] did not contain a UIN, update that item with the new UIN and grade. Else if that item had already been filled, continue down that linked list until the end. Create a new item with the given grade and UIN. The prev item should be pointing to this new item in order to continue the linked list.
      iii. **hash.searchGrade()** - First, create an index by hashing the UIN. Go to HashTable[index]. Cycle through that linked list until UIN of that item matches the UIN being searched for.

   (d) Analyze the algorithms according to assignment requirements.

      i. **hash.hash()** - $O(1)$
      ii. **hash.insert()** - $O(1)$
      iii. **hash.searchGrade()** - $O(n)$ where n is the number of collisions at that index.

3. A C++ organization and implementation of the problem solution

   (a) Provide a list and description of classes or interfaces used by a program such as classes used to implement the data structures or exceptions.

   class hash - private variables consisted of a constant table size value. The hashtable consisting of item nodes was also included. Public functions consisted of Hashing a UIN, inserting an grade and UIN to the hash table, calculating number of items at an index, printing table information, and searching for a grade for an associated UIN. A item struct was also used which consisted of a UIN, grade, and pointer to the next item.

   (b) Include in the report the class declarations from a header file (.h) and their implementation from a source file (.cpp).

   (c) Provide features of the C++ programming paradigms like Inheritance or Polymorphism in case of object oriented programming, or Templates in the case of generic programming used in your implementation.

   This program implemented one class and one structure. They are described above

4. A user guide description how to navigate your program with the instructions how to:

   (a) compile the program: specify the directory and file names, etc.

   To compile the program type ‘g++ -std=c++14 *.cpp -o run.out‘ to the terminal

   (b) run the program: specify the name of an executable file.

   To run the program type ‘./run.out‘

5. Specifications and description of input and output formats and files

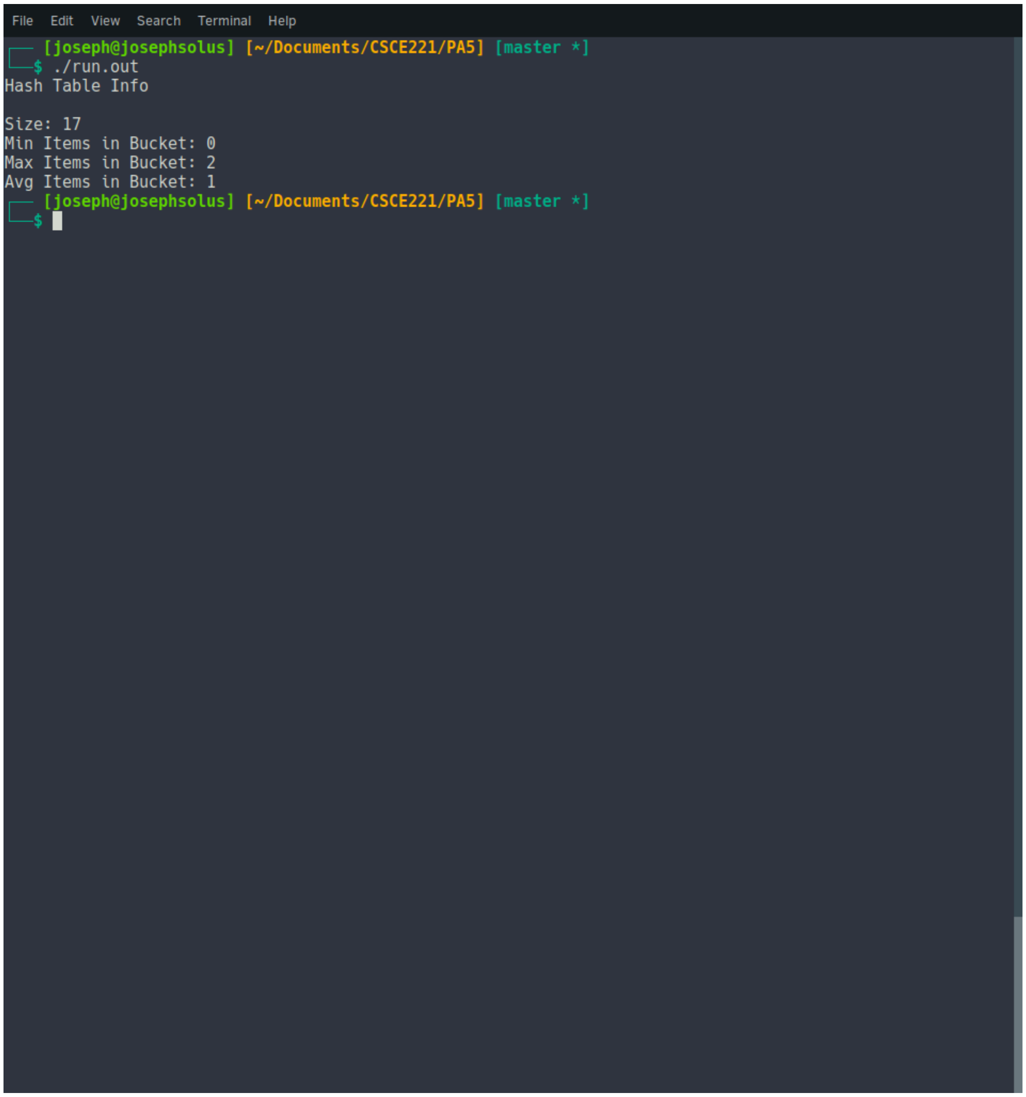
   (a) The type of files: keyboard, text files, etc (if applicable).

   The input consisted of two files. One is a list of UINs, names, and quiz grade. This file is called input.csv. The other input file consisted of UINs and names called roster.csv. The output file is output.csv which has Name, UIN, and quiz grade if the student took the quiz

   (b) A file input format: when a program requires a sequence of input items, specify the number of items per line or a line termination. Provide a sample of a required input format.

   Example: "Echo Wells,accumsan.neque@acmetus.com,800000079,94"

6. Test your program for correctness using valid, invalid, and random inputs (e.g., insertion of an item at the beginning, at the end, or at a random place into a sorted vector). Include evidence of your testing, such as an output file or screen shots with an input and the corresponding output.



```
┌─ [joseph@josephsolus] [~/Documents/CSCE221/PA5] [master *]
└──$ ./run.out
Hash Table Info

Size: 17
Min Items in Bucket: 0
Max Items in Bucket: 2
Avg Items in Bucket: 1
┌─ [joseph@josephsolus] [~/Documents/CSCE221/PA5] [master *]
└──$ 
```