25 points possible Due 4/21/2020

Programming Assignment 4

Statement of Work

1 **Overview**

In this programming assignment you will have the opportunity to study the effects of different hash functions on the efficiency of inserting data into a hash table.

2 **Requirements**

The student shall define, develop, document, prototype, test, and modify as required the software system.

2.1 **Functional Requirements**

2.1.1 The software shall include three functions each of which implements a different hash function. Hash functions may be those studied in class or adaptations of those functions.

2.1.2 The software shall include three functions each of which implements a different double hash function. One and only one of these double hash functions must be for linear probing, i.e. returns an increment value of 1.

2.1.3 Each hash function shall be tested using each of the double hash functions as a means of collision resolution. This means a total of nine tests. A list of 50 4-letter keys, each with some associated data will be provided for the testing.

2.1.4 The code to test these functions will be provided. Your job is just to write and test the hash and double hash functions.

2.1.5 Due to time constraints (from the assignment posting date being moved, the move online, etc.), there are some differences with this programming assignment compared to the previous ones:

1. No sprint report is required.
2. You should include the output from your program, in a text file, along with your source code.

3 **Deliverables**

These products shall be delivered to the instructor **electronically via Canvas** not later than **Tuesday, April 21, 2020**.

3.1 Program Source Files – The student shall provide fully tested electronic copies of the .cpp and .h files. **For this assignment, your main should be included**.

3.2 Output Text File – This should contain the output of a run of your program, which should include results of all experiments in its output. You can create this text file by copying and pasting the program output from the program console window, or, for those comfortable with the command line, via I/O redirection (e.g., `prog4 > P4.txt`)