**CS341 Software Engineering**

**Homework 7**

**Write the application described below in Java. Commit the source code (with documentation) to GitHub.**

**Submit the Git repository URL to Moodle.**

**Submit the following Word documents to Moodle**

* Testing summary.
* Completed final summary for your Personal Software Process: **PSP Form**

*Program Requirements, Class Diagrams, Testing, Personal Software Process*

 **Implement a Software Development Tool - Minimal Perfect Hash Function**

Begin this assignment by researching Hashing, Hash Tables, and Hash Maps.

Design and write Java console software that searches a source code file in Java looking for keywords, keeps statistics on the results of the search, and outputs these statistics to a file. All the key words must be stored in a file.

Design Issues:

1. Build a Java class for a minimal perfect hash table of keywords. This is one of the major components of your project.
2. You will need to open the keyword file and read each of the words into a vector. Once you have read all the keywords, you can start to build your hash table and define your hash function. The vector that all the keywords were initially read into will be the initial word list for the algorithm.
3. You may use the hash function below:

**h(word) = (length(word) + 26 \*(firstletter(word)) + g\*(lastletter(word))) % size**

**Term** **Description**

**h(word)** This will be the index into the hash table for the word.

**length(word)** The number of characters in the word.

**firstletter(word)** The character that is in the first position of the word.

**lastletter(word)** The character that is at the end of the word.

**size** Total number of keywords. It is also the number of elements in the hash table.

**g** Calculate the hashcode where g is the constant 31.

Note: Once a minimal perfect hash table is constructed, you can begin reading the source file and counting keywords.

1. A summary of all the statistics you must keep are presented in the following table:

Statistic Description

lines Total number of lines of code read. Blank lines and comment lines should not be counted in this total.

keywords This is the total number of keywords found in the source file.

Time Record the total time of your program, in milliseconds. This time should include the time to read the keywords, create the hash table, read the text file, and record all the statistics. Build a Timer class, with start and stop functionality. Stop timing immediately before writing the statistics to a file.

1. Produce a final summary of your Personal Software Process. See the **PSP HW Form**. This document requires you to record the size (number of lines of code) for your entire program, the time spent on design, coding, testing (in minutes), and a list of all defects your encountered and fixed (indicate the type/root cause). The final program should have all defects corrected. Retain this data for future use.