Ryan Ellis
COSC 320 --- 001 Lab 7 3/27/2025

Lab report: In this lab, we were given the header files for the hash function(s) and the hash table classes, along with a dictionary to input into the hash table and a sample text file to test the program. This exercise took me about an hour to finish, cumulatively.

Pre-lab:

1. Reviewed what we learned for hash table and hash function
2. Read and understand the header file d_hash.h, d_hashf.h, d_hiter.h,

Exercise 1:

```
//=============================================================
// Filename: lab_07.cpp
// Author: Ryan Ellis
// Creation Date: 3/25/2025
// Last Update: 3/25/2025
// Description: Main program that tests the usage of hash tables linked with buckets, inserting words from a dictionary data file,
// the program will ask the user for a text file to test whether words are spelled correctly according to the data file using the
// hash tables to check and return the findings.
// User Interface:
// Notes:
//=============================================================

#include <iostream>
#include "d_hash.h"
#include "d_hashf.h"
#include <fstream>

using namespace std;

ifstream& getWord(ifstream& fin, string& w);    //prototype for getWord

int main(){

   ifstream inputFile;        //input file, filename for user prompt, number of Buckets, hash function, and myhash object declarations
   string filename = "";
   int numBuckets = 1373;
   hFstring stringHash;
   myhash<string, hFstring> table(numBuckets, stringHash);
```

```
    inputFile.open("dict.dat");     //open input file

    if(inputFile){          //if file exists, input words from dictionary into hash table
        string line;
        while(getline(inputFile,line)){
            table.insert(line);
        }
    }

    inputFile.close();  //close the file

    cout<<"Enter the document name: ";      //prompt user for text filename to spell check
    cin>>filename;

    if(filename.find(".txt") == string::npos)       //if filename entered doesn't include file extension add it
        filename = filename + ".txt";

    inputFile.open(filename);       //open file

    if(inputFile){          //if file exists, run through inputFile and check if words are in hash table
        string line;
        cout<<"Misspelled words: \n"<<endl;
        while(getWord(inputFile, line)){
            if(table.find(line) == table.end())
                cout<< line<<endl;
        }

    }
    else{               //Error message if file cannot be opened
        cerr<<"Error opening the file.\n";

    }
    inputFile.close();

    return 0;
}

//extract a word from fin
ifstream& getWord(ifstream& fin, string& w)
{
        char c;

        w = "";// clear the string of characters

        while (fin.get(c) && !isalpha(c))
                ;       // do nothing. just ignore c

        // return on end-of-file
        if (fin.eof())
```

```
            return fin;

        // record first letter of the word
        w += tolower(c);

        // collect letters and digits
        while (fin.get(c) && (isalpha(c) || isdigit(c)))
            w += tolower(c);

        return fin;
}
```

Output:

```
ryan@ryan-MacBookPro:~/Documents/COSC 320/Labs/Lab 7/Lab 7$ make
make: 'prog' is up to date.
ryan@ryan-MacBookPro:~/Documents/COSC 320/Labs/Lab 7/Lab 7$ ./prog
Enter the document name: test.txt
Misspelled words:

tesst
ryan@ryan-MacBookPro:~/Documents/COSC 320/Labs/Lab 7/Lab 7$ ./prog
Enter the document name: test
Misspelled words:

tesst
ryan@ryan-MacBookPro:~/Documents/COSC 320/Labs/Lab 7/Lab 7$ ./prog
Enter the document name: spelltst
Misspelled words:

persn
types
documnt
makes
mistakes
mispells
teh
sords
ryan@ryan-MacBookPro:~/Documents/COSC 320/Labs/Lab 7/Lab 7$ ./prog
Enter the document name: test.txt
Misspelled words:

tesst
ryan@ryan-MacBookPro:~/Documents/COSC 320/Labs/Lab 7/Lab 7$
```