CSCI-2270
Assignment 1
Instructor: Boese

C++ Fundamentals

Objectives

- Read in command-line-args
- Read a file
- Loop through an array
- Split a string
- Create an array of struct
- Pass by reference
- Create a class with constructors/getters/setters
- Create header files with header guards

Problem Set

1. Read in a data file (the filename comes in via command-line arguments) with numbers on each line in the file and store these values into a sorted array. Create the array of size 100 and use an array sentinel to keep track of the actual number of entries *(will not exceed 100).* As you read each value in, place it into the correct position in the sorted array. Then prompt the user for a number, and report back to the user if it is in the list and if so, what position in the list.

Specifications:

1. Use getline to read in each line from the file.
2. Print out the name of the file you read in from command-line args.
3. Create a function named **insertIntoSortedArray** that takes the array, number of entries currently in the array, and new value to be inserted as parameters. Inside the function insert the new value into the correct position in the sorted array. Use the following signature:
   int insertIntoSortedArray(int myArray[ ], int numEntries, int newValue)
4. Print out the entire array each time after calling insertIntoSortedArray, displaying the values as a comma separated list.
5. Print out the total number of entries in the array after reading in the file

For example, Given a data file named **dat.txt** with the following values:

```
9
3
6
```

The output would be:

```
dat.txt
9
3,9
3,6,9
3
```

2. Read in a .csv data file where each line is structured as such: "username,gpa,age". Store these values into a struct with **username**, **gpa**, and **age** (string, float, int) and create a list as a vector of these structs. As you read in each line from the file, parse each value using stringstream and convert to the appropriate data type. Then call the **addUser** function to create the struct and add it to the vector. Then call the **printList** function to loop through and print out the list formatted as: "username [gpa] age: #" e.g., "elle [3.87] age: 12" .

   Specifications:
   a. Use the following struct declaration:

```
struct list
{
    string username;
    float gpa;
    int age;

};
```

   b. Use the following function signatures:

```
void addUser(vector<list> *gradeList, string name, float _gpa, int
_age)
void printList(const vector<list> gradeList)
```

   c. Use getline, stringstream, stoi, stof.

   d. Be sure to close your file when you are done.

3. Create a class named **Movie** that maintains information about the title (string), year (int), and rating (double). Create all accessors and mutators, and overload the constructor with a default constructor and one that takes all 3 values (title, year, rating). Make the default constructor initialize with the following default values: "unknown", 2016, 0.0. Write a driver program that creates three movies and tests all methods.

Submitting Your Code:
Log into Moodle and go to the Homework 1 link. It is set up in the quiz format. Follow the instructions on each question to submit all or parts of each

assignment question. *Note: there is no late period on assignments! If you miss the deadline or do not do well, you can sign up for an optional grading interview to get up to half the points missed back. There is also an optional extra credit assignment at the end of the semester you can use to replace one of your homework scores.*