Program #1
CS 163 Data Structures

Submit your program to the D2L Dropbox (sign on via d2l.pdx.edu)
Email a backup copy to karlaf@pdx.edu

LATE work will be accepted – but ONLY within a week of the original due date.
We do not accept late work beyond that. There are no exceptions.

Scope: When beginning with this project, the first thing to keep in mind is that we only have approximately 2 weeks to complete each assignment. Therefore, it is critical that you focus on a limited scope, with an emphasis on the class(es) and data structures. Your program **DOES** need to compile and run, but you won't be building complete applications. *You will be primarily graded on your use of classes, member functions, arguments, data structures, pointers and the efficiency of your code.* You will need a main, but it should focus on thoroughly testing out your class member functions rather than being a complete application program. Imagine that there is another software engineer who will be building the actual application software. Your job is to build the Abstract Data Type (ADT) and test it!

Therefore, focus on how to design classes that are well structured and efficient and on the required data structures. Limit the development of the application that uses the data structures. Of course, your user interface must be clear enough for us to test your program and we must be able to thoroughly test all features. **And, it is not appropriate to hard code in the tests - all tests should be interactive.** This first program of the term is an exercise in building, traversing, and destroying linear linked lists.

Background Information: My family is big into super heroes. Who has the best powers? Is it Wolverine with his amazing healing powers. Or can Jean Gray's telepathy win over a situation? Of course Professor X can just manipulate memories. Spider man might be the best in a city but wouldn't be able to use many of his powers in a smaller town or village. My personal favorite is the teleportation done by Nightcawler. How great is that? No more need for fossil fuel! My family and I have many discussions on which super hero power would be best used in a particular situation. My daughter has had arguments for hours with people – about Superman versus Thor. Thor always wins!

Programming Assignment: With program #1, you will be creating a program that will assist finding the best super hero for a given situation. The super hero powers will be read in from a file and stored in a linear linked list of dynamically allocated arrays. Each node will be about a particular type of power and each array index will be a super hero that exhibits this power and information about the super hero. The nodes should be sorted by power.

The user then can (a) add another super hero to the list of heroes with a particular power, (b) display all of the powers listed and the super heroes with those powers, (c) display just the super heroes that have a particular power, (d) remove a particular type of power (and all of the super heroes with that power) from the list, and (e) find the type of powers that would be best for the type of situation desired.

Super hero powers will be read in from a file. You will need two external files. One containing the super powers and another containing a description of each of the super heroes. Have semicolons separate the fields in the files:

1. Type of power (e.g. Healing)
2. A situation where this power could be most useful (e.g., Battle)
3. The number of super heroes that have this power (e.g., 3)
4. Super heroes that exhibit this power (e.g., Wolverine; Lobo; Claire Bennet)

You will need at least 3 structs for this program:

1. Power: Type of power and situation it would be useful in (e.g., type of power, situation)
2. Hero: Information about the super hero (e.g., name, description)
3. Node: Containing the type of power and a dynamically allocated array of Hero objects

Do not assume that the external data file is already sorted. In fact, make sure to test your program with data files that are not already ordered.

The two most important parts of this assignment are implementing the data structure and creating the ADT to manage the data structure. The following class is a suggested class interface for this first assignment:

```
class Hero_Match
{
    public:
        Hero_Match();
         ~Hero_Match ();
        int Add_Hero(char super_power[], const Hero &);
        int Load_Heros(char power_file_name[], char hero_file_name[]);
        int Remove_Power(char super_power[]);
        int Display();  //display all
        int Display(char super_power []);     //display all for a particular power
        int Find(char situation[], Power & found_power); //find the best power
    private:
          node * head;
};
```

Things you should know...as part of your program:

1) Do not use statically allocated arrays in your classes or structures. All memory must be dynamically allocated and kept to a minimum!

2) All data members in a class must be private

3) Never prompt and read from the user when inside a class member function

4) Never output error messages from a class member function

5) Global variables are not allowed in CS163

5) Do not use the String class! (use arrays of characters instead!)

6) Use modular design, separating the .h files from the .cpp files. Remember, .h files should contain the class header and any necessary prototypes. The .cpp files should contain function definitions. You must have at least 1 .h file and 2 .cpp files. **Never implement functions in your .h file! And, never "#include" .cpp files!**

7) Use the iostream library for all I/O; do not use stdio.h.

8) Make sure to define a constructor and destructor for your class. Your destructor must deallocate all dynamically allocated memory.

9) Remember that 20% of each program's grade is based on a written discussion of the design. *Take a look at the style sheet which gives instruction on the topics that your write-up needs to cover.*