Programming Assignment #4
CS 162: Introduction to Computer Science

Submit your assignment to the D2L Dropbox

*** Your program will need to have at least
TWO .cpp files, ONE .h file ***

The purpose of the 4th program is to implement the new concepts learned which include (a) pointers, (b) dynamically allocated arrays, and (c) classes. The purpose of the 5th program is to incorporate linear linked lists. Our goal with both assignments is to continue to create programs with a small main function that delegates to a series of functions where the real work takes place.

In this programming assignment, you are **not** allowed to use global variables. You are allowed (as usual) to use the cstring library (e.g., strlen, strcpy, and strcmp). Limit your main (and all functions) to no more than 30 statements of code (for executable statements… *not counting variable definitions, blank lines, lines with just curly brackets, or comments*).

Program Assignment:
I think the hardest thing about being remote right now is trying to figure out how to connect with everyone. There are zoom links, google meet links, email, slack messages, D2L messages, text messages, facebook, and so many more. I am surrounded right now by six screens and two ipads and all are lit up with different screens of events that are taking place or conversations that are happening. It is really easy for things to fall through the cracks. Did we forget to go to this link or go to that meeting?

For programs #4 and 5 we will be working on building a list class to manage all of the different links and/or appointments that we need to keep track of in order to be successful with a course (for example).

For each individual link we want to keep track of the following. This may be in a structure or a class (we recommend a structure to start!):

1. The name of the link (e.g., Google Meet)
2. The link itself (e.g., meet.google.com/ipb-pbrb-wmm?blabla)
3. The reason for the link or appointment (e.g., proficiency demo)
4. The amount of time I need to allocate for this activity (e.g., 45 minutes)
5. If it involves other information (like do I also need to be logged into D2L?)
6. Other important information (e.g., if it was a group project, then who are the group members)

Then, what is most interesting is to hold a list of all of the different links and/or appointments in a class; I refer to the previously discussed link information as: Link_info, but you may name it anything you would like. The minimum set of class member functions are the following:

```
class Important_Links
{
    public:
                Important_Links();          //constructor
                ~Important_Links();         //destructor
                void Add( Link_info & to_add);
                void Display_all();
                void Display_benefit(char name[]); //display just  links of this type
   private:
      //Put the information about an array of Link_info structures here
      //For this assignment, this should be a dynamically allocated array of
      //Link_info and an integer count
};
```

Make sure to use *dynamically allocated arrays in Program #4 (they should be used whenever we actually don't know the size of an item when we are writing our program)!*

Then, in Program #5, we will make a linear linked list of Important_Links. This means that each node needs an Important_Links object and a next pointer. This will allow us to have a list of different courses each with their own important links!! ***This linear linked list does not need to be in a class.*** *It can be managed by a function (such as main) to build a linear linked list, populate it with links for your different classes, and then destroy it. Remember, all memory dynamically allocated (with new) needs a corresponding use of delete to deallocate it.*

******You are always welcome to do more! But, really focus on making general purpose functions that can be re-used. Anytime you have code that has already existed elsewhere in your program (such as to error check input or give the user another chance), write a function instead!***

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

1. Make sure to prompt the user for any input requested. Make sure it is clear from your prompts what the user is expected to do.
2. The program should continue until the user wants to quit. Allow them to continue until they are done.
3. You may not use any global variables in this program!
4. You may not use the string class – instead use arrays of characters. Remember the cstring library is fine!
5. You will need to read using the 3 argument version get for the key features and location so make sure to use ignore to remove extra characters (and the delimiter) from the input stream.
6. Error check for faulty input. You can assume the grader will enter in the appropriate data type if your prompts are clear.
7. FUNCTIONS are required for this assignment.
8. Make sure to put your name in your program

It is important when working with multiple files to first:
(a) create a backup of your files and then
(b) create an archive of the files using tar or zip.

Follow the instructions from Lab #6 closely.