

Justin Greever  
CS162  
Program 4 – Favorite Activities stored using nodes

#### //Classes

I plan on creating a class specifically for the activities required input from the user. This will consist of private char arrays: activity[], location[], timeOfYear[], description[], thoughts[], supplies[], specificDates[]. The public part of the class will contain: read(), show(), find(), save(). These will interact with the functions created later on.

#### //Prototypes

There will be a prototype load\_from\_file that will be responsible for checking if a file containing the information already exists, and if so, load that file into memory. Prototype save\_to\_file will copy the information from the nodes to a text file on the hard drive so the user doesn't lose the data when they quit the application. Prototype to\_display will allow the user to display the data they just entered, or after performing a search, display the data found. Prototype to\_read is used to grab input from the user and insert it into the linked list. This will also be used as the starting point for the menu of choices for the user to choose from. Prototype search\_activity will traverse the linked list and stop when it has found a matching value to the user's input. This will return the information for the user to view. Prototype display\_all will print all data that is currently stored in the linked list. If no data found, it will output "No data found" and return to the main menu.

#### //Functions

load\_from\_file(), this will load data from the text file and insert it into the linked list using the class created earlier. If this function loads data, it will return the number of entries to the user, if no data found, it will simply print "no data found".

save\_to\_file(), this will copy the contents from the linked list using the class created and save the data into a text file. It will use a hashtag as a separator (#). If the file doesn't exist, this function will create the file for you automatically.

to\_display(), this function takes the user's input that they just entered and displays it back to them using the class design. This function is also used when the user searches for data later on and wants to display it. It prints out the information, and doesn't pass anything back to main when it's done.

to\_read(), this takes the user's input and arranges it via the class design and saves the information into the linked list. If there is already information in the list, it will traverse to the last spot and append the information there.

search\_activities(), this will ask the user for their search criteria, most likely the activity name at first, and then traverses the linked list to search for that activity name. Once found, it will output the entire entry to the user. This doesn't pass any arguments back, it's simply to search and display.

display\_all(), this starts at the head and traverses the linked list, displaying the entire list of activities and all their associated information. This returns nothing back to main.

#### //Main

Within main, there will be a char variable to hold the response (Y/N) value from the user, a char variable for the menu entry that the user selects, and an int variable to retain the activity count from a file being loaded. The main menu will be displayed upon starting the application. Its entries are:

L – load, A – add, S – search, D – display, Q – quit

If a user enters anything other than one of these, it will simply state invalid entry and repeat the menu options for the user. Each function when it's finished will bring the user back to the main menu so they can make a selection again. Q simply does a return 0; if all other conditions are met.