Kyle Knudson

CSCI 591

Cs301188

Section 1

02/27/18

Project 6

Project 6 - Design Document

Introduction:

This program involves the creation and utilization of linked lists. A menu driven system was constructed in this program to allow the user to have control over manipulating a linked list. The user is allowed to continue to make alterations to a list or make queries about a list until the user quits the program. Concepts from object-oriented programming have also been implemented by creating a list class and using this class to perform the linked list manipulations. This program also incorporates the concept of recursion into the list class by using functions that access the list as recursive functions. This allows for the client program to remain unchanged but the functionality of the class is altered in how it traverses through the data structure. As a programming exercise this illustrates how changing the back end functionality of a program does not necessarily have any impact on the client program.

Data Structures:

In this program, the abstract data type that is used is the ordered list data type. This data type is where the list of integers is stored so that they are able to be manipulated by the List class’s functions.

Functions:

List() – this function is the constructor for the list object.

List(const List & source) – this function is the copy constructor and is used to copy a list into another list.

void make\_empty() – This function removes the contents of a list so that the list is empty.

void insert(int entry) – This function inserts an entry into the linked list in the correct order.

void remove (int target) – This function removes a target value from the linked list and connects the other values in the correct order.

void operator = (List source) – This function overrides the assignment operator and assigns the valus from the first list ot another value.

bool is\_empty() – This function is a Boolean function that returns true if the function is empty and returns false if the first pointer value is not equal to NULL.

int length(Node \*) – This function returns the integer value of the linked list by keeping a counter as it moves a pointer down the list until it reaches NULL.

bool present(int target) – This function is a Boolean function that returns true if a target value is in the linked list and false if the value is not in the list.

int kth (int k) – This function returns the kth value in the linked list. This is done by using a loop and moving the pointer down the list a specific number of values.

friend ostream & operator << (ostream & out, List l) – This function overrides the output operator so that a list is able to be sent to an output stream.

Node \* get\_node(int entry, Node \* link) – This function gets the specific node in a list.

print() – This function prints out the contents of each node as it moves down the list.

There is also private recursive functions for kth(), present(), length(), remove(), insert(), and make\_empty(). All of the recursive functions begin with r\_followed by one of these functions. These functions actually perform the functions while the public version simply calls the recursive one.

The Main program:

The main program for this project is relatively simple. It consists of a single while loop that continues to operate the menu until the user selects the option to quit out of the program. The menu allows the user the options of manipulating lists until the user is exits from the program. The loop consists of a switch statement that controls the menu system depending on what the user inputs. The user will continue to be allowed to do operations on the list until the quit option is selected by the user. The option that the user selects calls the public functions which are used to call the recursive functions in the list class. The manipulations are done on a single linked list in this program. There is also error checking to see if the input is upper case or lower case. Both options will work if entered.