CS 2401 – Spring 20212

Project 3 – Linked Lists

Due: 11:59 PM Sunday, March 6th

For this project, you will be completing a linked list container that will store information about a student's pending homework assignments. **This list will always be kept in order by the date that an assignment is due.**

To get us started on this project, the *Date* class that we used in the last two projects has been altered so that it can now have a date object set to today, using the computer's internal clock.

In the same file (to keep the number of files that we are working with smaller) there is a new class called *DateTime*, which records the hour and minute when something happened as well as a date.

Using these two, a class called *Assignment* is defined, which records the name of an assignment (a string), the course the assignment is for (a string), the date/time the assignment was entered (a *DateTime*) and the date/time when the assignment is due (a *DateTime*). The overloaded comparison operators for the *Assignment* class are based on the date/time an assignment is due, so if assignment1 < assignment2 it means that assignment1 is due before assignment2, disregarding name, course, and time the assignment was entered.

You should begin the project by downloading:

datetime.h
datetime.cc
assignment.h
assignment.cc
node.h (has the assignment as its data type)
main.cc

Your task is to create is to create the header and implementation files for the Planner class. This class will store a list of assignments that the student needs to complete. This list is stored in the form of a linked list of nodes. (I have given you the node class you are to use.) The list is **always** kept in order by the assignment due date, but the people who put the things into the list may not be recording them in that order. This means that when a new assignment is added, you will need to search through the list to put it in the right place. All comparison operators are overloaded for the Assignment class, to facilitate this.

The Planner class will offer the application programmer the following options:

1. Add a new Assignment to the list. (It will be inserted in order.)
2. View the entire list of Assignment, in the order they are stored.
3. Remove an assignment that has been completed. This will use the find and remove functions, both of which will take a string which is the name of the assignment.
4. Learn the number of assignments that are waiting to be completed.
5. See the amount of time you have to complete the assignment that is due soonest. (This assignment should be at the front of the list.)
6. See the average amount of time that assignments have been sitting in the list since they were entered.
7. Identify the assignment that has been in the list the longest.
8. Identify the assignment that was added to the list most recently.
9. See all assignments that must be completed by a date entered by the user.

The main that I have given you has a menu for all these options, and you should be able discern the names and parameters for the required functions from the calls that you see in the main.

You will also need to have a `load` function, and a `save` function that saves simply the information about the assignments, not the size of the list. The data file should be formatted:

assignment name
course
date the assignment was entered
time the assignment was entered
date the assignment is due
time the assignment is due

Since linked lists are built using dynamic memory, your Planner class will be “holding” dynamic memory and that means that you will need the implementation of the Big 3, and I have built something that will let me check your copy constructor.

There are some useful functions already included in the datetime.cc and assignment.cc files that can help you with calculating times so do not try to recreate all of that. There are also overloaded >> and << operators for the Assignment class, so you should use those as well.

For this project you should submit the `planner.h` and `planner.cc` files that you have created.