ECS 40 **Program #6 (50 points, 2.5 hours)** **Spring 2017**

Due: Friday, May 19th at 11:59 pm in p6 of cs40 using handin.
New concepts: polymorphism, templates, vectors.
File names (exactly): authors.csv, main.cpp, calendar.cpp, calendar.h, day.cpp, day.h, appt.cpp, appt.h, time.cpp, time.h, DayOfWeek.cpp, DayOfWeek.h, linkedlist.cpp, linkedlist.h, weeklyappt.cpp, weeklyappt.h, vector.cpp, vector.h, and Makefile.

This is an extension of program #5. You are to modify program #5 so that: 1) the year of an appointment is now displayed; 2) there is a templated Vector class that resizes its array as needed, and use this class to instantiate the days array in Calendar; 3) the LinkedList and ListNode classes are now template classes, and modify the Day class accordingly; and r) there is a WeeklyAppointment class that is derived from the Appointment class that indicates that an appointment happens at the same time and place on multiple days.

You will find my executable, DOW.dat, and appts.csv in ~ssdavis/40/p6. You are encouraged to use my source code from program #5 as the basis for your program #6

Further specifications and suggestions:

1. (10 minutes) Modify the DayOfWeek insertion operator so that the year of the appointment is also displayed.
2. (30 minutes) Create a templated Vector class, and make Calendar::days a Vector of Day.
   - 2.1. In the Makefile, make calendar.cpp dependent on both vector.cpp, and vector.h. Do NOT write a rule to create vector.o!! For Netbeans and other IDEs, do not add the files to the project, but you can place them in the "Important Files" category—just so they are not explicitly compiled.
   - 2.2. The Vector class has a T* named array, an int named count, and an int named size.
   - 2.3. Remember to add `#include "vector.cpp"` at the bottom of vector.h.
   - 2.4. A good way to develop is to add a Vector<Day> days2; to your Calendar class. It won't be used yet, but you can regularly compile to see if your code has any syntax errors without waiting until you have completed the whole class.
   - 2.5. To make your code work, the class has a constructor, destructor, getCount(), a const operator[], and a non-const operator[], and an overloaded += operator.
     - 2.5.1. The constructor should take its initial size as its only parameter.
     - 2.5.2. The overloaded += relies on the overloaded < operator of , and returns the position of the inserted object. The method incorporates the code from the Calendar::resize() and Calendar::findDate() functions. "days" should be replaced with "array," and "dayTemp" replaced with "object" so all of the Vector code is generic.
   - 2.6. Once everything compiles, change calendar.h to reflect the new type for days, and eliminate its count and size data. If all goes well, the only things you will need to change in calendar.cpp are:
     - 2.6.1. Initializing days with 30 in the Calendar constructor
     - 2.6.2. Use days.getCount() instead of count.
     - 2.6.3. Eliminating the days array size maintenance stuff throughout the code. Since the Calendar class no longer contains count and size, the compiler will tell you where you have code that needs to be re-worked.
3. (25 minutes) Changing LinkedList and ListNode to template classes.
   - 3.1. Since it will be a template class you must change the Makefile. Remove the rule to create linkedlist.o, add linked.cpp to the dependency list of day.o, and remove any mention of linkedlist.o from the Makefile. In Netbeans and other IDEs, you must remove (but not delete) linkedlist.h and linkedlist.cpp from the Project. You can add them to "Important Files" so you can easily access them.
   - 3.2. To allow for polymorphism and keep approaching the changes simpler, we will now have the LinkedList class assume that what it is storing is a pointer. Thus the declaration in day.h will be a <Appointment*>. This will mean that the ListNode class simply contains a T, not a T*. The LinkedList methods that used to return an Appointment*, will now return a T.
   - 3.3. You will need to add a virtual clone() function to Appointment and WeeklyAppointment that returns a pointer to a new object based on a call to the copy constructor of the respective class, e.g. new Appoiintment(*this). This will require you to write a copy constructor for WeeklyAppointment. clone() will be called instead of simply "new T" in the LinkedList::operator= operator.
   - 3.4. You should eliminate the non-const operator[].
   - 3.5. Add #include "linkedlist.cpp" at the bottom of linkedlist.h.

3.6. Adding "template <typename T> above the class declarations in linkedlist.h, and above every function in linkedlist.cpp.
3.7. Use global search and replace to place a " <T>" after both "List" and "ListNode" in linkedlist.cpp. You will have to revert the constructors and destructor function names, but it is worth saving the time on the other functions, particularly all those ListNode *ptr…
3.8. Use global search and replace to replace the variable name "appointment" with "object," and to replace "Appointment" with "T".
3.9. Page 665 in the textbook provides good guidelines for writing the various forms of friendship. Note that the operator<< function must be forward declared, and the prototype within the class definitions has a <T> after the function name.
3.10. The actual calls in day.cpp should work without any changes at all!

4. WeeklyAppointments have the additional data members of char series[8], and int seriesCount.
4.1. Note that the Appointment class members MUST remain private.
4.2. The series array is used to store the days of every week upon which the appointments occur.
4.3. There will be a separate WeeklyAppointment object each of its days. For example, a WeeklyAppointment with a seriesCount of 13 will have copies in 12 days after the original date.
4.4. Appts*.csv file format
4.4.1. After the date, a new field indicates whether the appointment is for a single day (0), or a WeeklyAppointments (1).
4.4.2. Lines with WeeklyAppointments end with their days of week and count. You may assume that the date provided is on one of those listed in the series. Lines without WeeklyAppointments just end with two commas.
4.5. Reading the file (75 minutes)
4.5.1. (30 minutes) You will need to write virtual read() functions for each of the Appointment classes to read their lines properly. Both read functions take an istream reference as their only parameter. It makes sense to have the WeeklyAppointment::read call the Appointment::read function to process the part of the line that is used by all Appointments. However this means that the Day extraction operator, must read the rest of the empty line for a regular Appointment. Once you have written the virtual reads, you should be able to run your program with appts4.csv. Test the date 8/8/2012 to see if the first appointment in that file is processed correctly. It should just look like a normal appointment at this point and have three appointments.
4.5.2. (45 minutes) The Day extraction operator will now return a const pointer to an Appointment instead of an istream reference. This will permit Calendar to create additional appointments when the Appointment is a WeeklyAppointment. Hint:use dynamic_cast to decide whether to call a brand new function named createSeries().
4.5.2.1. Add accessor methods to the WeeklyAppointment class for each of its pieces of data.
4.5.2.2. Add a DayOfWeek::operator==(char c) method that compares 'M', 'T', 'W', 'R', 'F', 'S', 'U' with its dayName. This will make storing WeeklyAppointments in succeeding days relatively easy.
4.5.2.3. Add a Day::operator+= (const WeeklyAppointment&) that creates a new WeeklyAppointment as the value for the call to LinkedList::+=. You will have to write a copy constructor for WeeklyAppointment.
4.5.2.4. Calendar::createSeries() takes a WeeklyAppointment, day, month, and year. I spent an hour debugging learning that you MUST pass the WeeklyAppointment by value because the pointer supplied by the Day extraction operator becomes invalid if the days array resizes while createSeries() is running! Passing by value doesn't require any new work on your part since you already wrote the copy constructor to get the Day::operator+= to work. By incrementing the day (and occasionally month and year) in a loop, you can use DOW.dat to find valid days into which you should insert the weeklyAppointment. Note that you should not need to add any other new methods to DayOfWeek, nor Appontment to accomplish this task.
4.5.2.5. Your program should run now, and you can check to see if 8/20/2012 has two appointments in it.
4.6. Displaying a WeeklyAppointment (10 minutes)
4.6.1. When printed to the screen, WeeklyAppointments print their series and count after the normal information.
4.6.2. You must continue to use the operator<< for printing both Appointments and WeeklyAppointments.
4.6.2.1. Note that ostream operators cannot be virtual because they are friends and not member methods, but they may call virtual methods named write().
4.6.2.2. Again, it makes sense (and is unavoidable since Appointments data is private) to have the WeeklyAppointment::write() call the Appointment::write(). I used setw(26) for location.

4.6.2.3. Note that the LinkedList insertion operator, and Date::subjectSearch() will now have to append an endl to its output, since the Appointment::write() cannot.

```
[ssdavis@lect1 SeansSrc]$ head appts4.csv
Date,Class,Subject,Start Time,End Time,Location,Series,Count
8/8/2012,1,Psychiatry,10:10:00 AM,12:50:00 PM,6 Olson,WF,10
3/9/2004,0,Psychiatry,9:00:00 AM,10:45:00 AM,1003 Geidt,,
1/21/2000,0,German,6:10:00 AM,8:55:00 AM,1100 Social Science,,
1/17/1995,0,Linguistics,5:50:00 PM,7:00:00 PM,1003 Geidt,,
5/10/2000,1,Mathematics,5:40:00 PM,6:45:00 PM,176 Everson,WRF,8
1/11/2005,0,Statistics,2:40:00 PM,4:20:00 PM,234 Wellman,,
4/16/2001,0,Music,5:50:00 PM,8:10:00 PM,100 Hunt,,
12/28/1997,0,Spanish,1:20:00 PM,2:45:00 PM,194 Young,,
10/1/2010,0,Music,5:25:00 PM,6:55:00 PM,2205 Haring,,
[ssdavis@lect1 SeansSrc]$

[ssdavis@lect1 SeansSrc]$ calendar.out appts4.csv
Calendar Menu
0. Done
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Print appointment count.

Your choice >> 1

Please enter the month, day, and year (mm/dd/yyyy) >> 8/15/2012
Start End   Subject      Location
06:30 06:50 Surgery      194 Young                WTRSF   (17)
10:10 12:50 Psychiatry   6 Olson                  WF      (10)
14:35 15:45 Geology      234 Wellman              WTR     (16)

Calendar Menu
0. Done
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Print appointment count.

Your choice >> 2
Please enter the subject >> Psychiatry
Date                          Start End   Subject      Location
Monday, May 17, 1993          12:35 15:05 Psychiatry   1001 Geidt
Saturday, October 9, 1993     07:15 09:55 Psychiatry   106 Wellman
Friday, January 28, 1994      09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Saturday, January 29, 1994    09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Sunday, January 30, 1994      09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Monday, January 31, 1994      09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Friday, February 4, 1994      09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Saturday, February 5, 1994    09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Sunday, February 6, 1994      09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Monday, February 7, 1994      09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Friday, February 11, 1994     09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Saturday, February 12, 1994   09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Sunday, February 13, 1994     09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Monday, February 14, 1994     09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Friday, February 18, 1994     09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Saturday, February 19, 1994   09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Sunday, February 20, 1994     09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Monday, February 21, 1994     09:10 11:35 Psychiatry   1003 Geidt               FSUM    (16)
Saturday, July 1, 2000        11:20 11:35 Psychiatry   194 Young
Saturday, March 8, 2003       10:30 11:55 Psychiatry   198 Young
Monday, March 10, 2003        14:00 15:05 Psychiatry   1100 Social Science
Tuesday, March 9, 2004        09:00 10:45 Psychiatry   1003 Geidt
Saturday, June 5, 2004        16:00 16:20 Psychiatry   1100 Social Science
Thursday, March 24, 2005      14:40 15:50 Psychiatry   123 Sciences Lecture Hall
Saturday, April 15, 2006      12:05 12:40 Psychiatry   234 Wellman
Wednesday, November 1, 2006   06:55 07:35 Psychiatry   6 Welman                 WFS     (7)
Friday, November 3, 2006      06:55 07:35 Psychiatry   6 Welman                 WFS     (7)
```

```
Saturday, November 4, 2006    06:55 07:35 Psychiatry   6 Welman                 WFS     (7)
Wednesday, November 8, 2006   06:55 07:35 Psychiatry   6 Welman                 WFS     (7)
Friday, November 10, 2006     06:55 07:35 Psychiatry   6 Welman                 WFS     (7)
Saturday, November 11, 2006   06:55 07:35 Psychiatry   6 Welman                 WFS     (7)
Wednesday, November 15, 2006  06:55 07:35 Psychiatry   6 Welman                 WFS     (7)
Monday, September 27, 2010    10:30 10:50 Psychiatry   66 Roessler
Wednesday, August 8, 2012     10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Wednesday, August 8, 2012     10:20 13:10 Psychiatry   234 Wellman
Friday, August 10, 2012       10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Wednesday, August 15, 2012    10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Friday, August 17, 2012       10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Wednesday, August 22, 2012    10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Friday, August 24, 2012       10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Wednesday, August 29, 2012    10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Friday, August 31, 2012       10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Wednesday, September 5, 2012  10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Friday, September 7, 2012     10:10 12:50 Psychiatry   6 Olson                  WF      (10)
Sunday, January 12, 2014      08:25 08:40 Psychiatry   3 Kleiber
Tuesday, March 28, 2017       07:05 09:30 Psychiatry   2 Wellman

Calendar Menu
0. Done
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Print appointment count.

Your choice >> 0
[ssdavis@lect1 SeansSrc]$
```