dDue: Wednesday, May 3rd at 10:59 pm in p4 of cs40 using handin.
New concepts: version control using bitbucket.org, constructors, overloaded assignment operators, destructors, random access binary files, and static variables,.
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, Commit.txt, and Makefile.

Creating a P4 Repository on bitbuck.org

Before starting the assignment, you will need to install git, and set-up and account and repository on bitbucket.org. I have chosen bitbucket.com over github.com because bitbucket.com has free private repositories. When repositories are private, other people cannot copy your code, and thus avoids the issue of plagiarism charges in ECS classes.

Installing Git

The tutorial at https://git-scm.com/book/en/v2/Getting-Started-Installing-Git details how to install Git on Linux, Windows, and Macs.

For Windows, I downloaded from http://git-scm.com/download/win, and accepted all of the default options.
For Mac, I downloaded from http://git-scm.com/download/mac.

Creating a BitBucket account
For Windows, there is a tutorial at: https://www.youtube.com/watch?v=5dU4f9vfeoM

1. Go to bitbucket.org, and select "Get started for free."
2. Fill in all the required, and submit it. Use your UCD e-mail account so you get the Academic License. This permits an unlimited number of collaborators instead of just five.
3. Wait for the e-mail bitbucket sends you, and click "Verify my e-mail address" in the e-mail sent you by bitbucket
4. Back in the browser, choose your username.
5. The Dashboard should appear. Click "Create a repository"
   5.1. Name the repository P4
   5.2. Make sure that "This is a private repository" is checked, and the repository type is Git.
   5.3. Click "Create a repository" on this window.
6. The Overview window should now appear. In the upper right hand corner there should be a URL of this particular repository. Copy the URL. It would be something like: https://ssdavis@bitbucket.org/ssdavis/p4.git
7. If you have a partner, then have that partner create a BitBucket account on another browser window. Now you can select Teams->Create a team to add them to your team for this project.

Using Netbeans with bitbucket

1. Create a new C/C++ application in Netbeans, and copy all of your (or my) P3 code to the directory of the project.
2. Add the .cpp to the Source File section of the project, and add the header files to the Header File section.
3. Right click on the project in the Project window, and select Versioning->Initialize Git Repository
   3.1. A new window opens asking you to determine where the repository should be created. Accept the default.
4. Go to Team->Remote->Clone
   4.1. Paste the URL from when you created the bitbucket repository into "Specify Git Repository Locaton. You will then be asked for your bitbucket User and Password. Enter them, and check the Save Password box.
   4.2. If the Specify Destination Folder looks correct (and it should be) click Finish.
   4.3. When the Clone Completed dialog appears just click the Close button.
5. Select Team->Commit. Write something appropriate in the Commit message (like original p3) in the commit message. Click the Commit button.
6. Select Team->Remote->Push…
   6.1. Simply click the Next button.
   6.2. In the Local Branches window, place a check mark next to the master, and then click the Next button.
   6.3. Click Yes when confronted with the question of Setting up remote tracking
7. You use Commit to save the current version locally.
8. You use Push to save the current version remotely on bitbucket.
9. You can learn more about using Netbeans with git at https://netbeans.org/kb/docs/ide/git.html#github

10. Go to Team->Branch/Tag->Create Branch. Name the branch constructors.

Editing P4

This is an extension of program #3. For this assignment you will: 1) convert create() or read() methods to constructors; 2) add an overloaded assignment operator to the Day class; 3) convert destroy methods to destructors; 4) read the names of the day of the week and month for any day from a binary file, 5) add an appointment, and 6) use a static variable to keep track of how many appointments exist at any given time.

You should be able to compile after each minor step, and run properly after completing each major step.

1. Convert create() and some read() to constructors by changing prototypes, implementation, and calls. Note how the filenames change to blue in the Project window when you have edited them.
   - 1.1. Make Calendar::create() a default constructor. There is no longer a need for a call to create() in main().
   - 1.2. Make Day::create() a standard constructor. All four variables should be initialized in the initialization list.
     - 1.2.1. In dateSearch(), declare dayTemp after the call to getDate().
     - 1.2.2. In readFile(), move the declaration of dayTemp after the three calls to strtok() to read the date.
     - 1.2.3. Because you are creating an array of Day, and you now have an explicit constructor, you must also create a default constructor for Day that does nothing.
   - 1.3. Create a default constructor for Appointment that sets the location and subject to NULL.
   - 1.4. Write a copy constructor for Appointment. Don't simply copy the values of location or subject! You must allocate new space for them. Copy the times in the initialization list. This is not called yet, but will be in the next step.
   - 1.5. After ensuring that everything works correctly, select the project title, and then click Team-> Commit. Change the commit message to "constructors"
2. Add an overloaded assignment operator to Day that creates new appointments relying on the copy constructor of Appointment. This will be necessary when we add the destructor because we will be deleting an array of Day in resize(), and the Day destructor will delete all of the appointments when the old, smaller array is deleted.
3. Convert the destroy() methods to destructors. Eliminate all calls to destroy. You can eliminate the for loop in ~Calendar since when an array of objects is deleted, the destructor of the each element is called.
   - 3.1. After ensuring that everything works correctly, select the project title, and then click Team-> Commit. Change the commit message to "destructors"
4. Make use of the DayOfWeek class in Day::subjectSearch() method to print the information about day as shown.
   - 4.1. The DOW.dat file contains binary representations of DayOfWeek objects starting from January 1, 1990.
   - 4.2. The DayOfWeek class must contain char monthName[10], int month, int day, int year, and char dayName[10]; in that order.
   - 4.3. DOW.dat contains 31 days for every month of every year. So there are 12 * 31 = 372 DayOfWeek objects in the file for each year. Days that do not exist have a dayName of "Invalid."
   - 4.4. The DayOfWeek::read() method takes three ints of a specific day. It will open DOW.dat, read only the DayOfWeek corresponding to that date.
     - 4.4.1. If you are having troubles with reading correctly, I suggest you try reading 1/1/1990, i.e. the first day in DOW.dat. It was a Monday. If that doesn't read properly, then you have probably declared the data variables of DayOfWeek improperly.
     - 4.4.2. You must use seekg() to locate the proper DayOfWeek in the file, and not any form of a loop.
   - 4.5. The DayOfWeek::print() methods prints the date including the name of the month, and the name of the day of the week.
     - 4.5.1. Hint: I used strcpy(), strcat(), and integer/char analysis to create one string for the entire date. Try to work this out for yourself. I hope that no one on Piazza will give away the solution.
   - 4.6. There can be only one DayOfWeek object within the whole program; there are no arrays of DayOfWeek.
   - 4.7. After ensuring that everything works correctly, select the project title, and then click Team-> Commit. Change the commit message to "Days of week"
5. Add a menu item that permits the user to add an appointment. If the date does not yet exist in the days array, then insert it. Different classes should handle requesting and checking their data from the user. However, Calendar::addAppointment() will need to call getDate(). You may assume that the format of the times entered will be valid. Otherwise, the design is up to you. Since there is a lot of overlap with previous tasks, you may wish to create auxiliary functions that can be called from two different aspects of the program. Do not change appts.csv!

5.1. After ensuring that everything works correctly, select the project title, and then click Team-> Commit. Change the commit message to "Add appointment"

6. Add a menu it that will print out the number of appointments currently in the entire calendar. This must rely on a static count variable of the Appointment class, and not call any function other than getCount() of the Appointment class.
   6.1. After ensuring that everything works correctly, select the project title, and then click Team-> Commit. Change the commit message to "Add appointment"

You will find my executable, DOW.dat, and appts.csv in ~ssdavis/40/p4. You are encouraged to use my source code from program #3 as the basis for your program #4.

Creating Commit.txt

1. Open a command line interface that has git in its path. When you download git from git-scm.com, they should have installed Git-Bash and/or Git-CMD. Opening either will do.
2. Change directories to where your local repository is located. When you use "`ls -a`", you should see "`.git`".
3. Type: "`git init`". This should load git with your repository.
4. Type: "`git log`". This should show all of your commits.
5. Redirect `git log` into Commit.txt.

```
[ssdavis@lect1 SeansSrc]$ calendar.out
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) >> 4/26/2004
Start End   Subject      Location
06:45 08:50 History      1002 Geidt
06:55 09:20 Spanish      6 Olson
10:05 11:15 Music        123 Sciences Lecture Hall
14:00 14:45 Classics     6 Welman

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 >> English
Date                        Start End   Subject      Location
Friday, November 25         12:05 13:45 English      1100 Social Science
Monday, April 16            13:00 14:40 English      1100 Social Science
Monday, February 6          07:40 08:45 English      66 Roessler
Invalid, February 31        06:50 07:05 English      1100 Social Science
Monday, September 12        07:05 07:15 English      206 Olson
Thursday, August 16         11:55 13:35 English      100 Hunt
Thursday, August 16         12:40 14:55 English      2205 Haring
Thursday, August 16         15:55 16:10 English      26 Wellman
Tuesday, May 26             16:05 17:20 English      234 Wellman
```

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

Your choice >> 4
Appointment count: 341

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

Your choice >> 3

Please enter the month, day, and year (mm/dd/yyyy) >> 1/1/1990
Subject >> New subject
Location >> 1001 Giedt
Start time (hh:mm) >> 10:12
End time (hh:mm) >> 11:23
Calendar Menu
0. Done
1. Search for date.
2. Search for subject.
3. Add an appointment.
4. Print appointment count.

Your choice >> 4
Appointment count: 342

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]$
```