**Assignment Book**

*Data Structures, Assignment 1A*

*6/28/2016*

# team members: Gulnoza Khakimova, Santiago Alvarez, Rebecca Peabody



*Summary*: This project models an assignment book suitable for student use in adding and completing assignments, allowing users to specify an assigned date, due date, and description for each assignment. From the primary interface users interact with a menu that gives access to the assignment book’s features including adding an assignment, finding and displaying an assignment, displaying the entire contents of the assignment book, editing an assignment, completing an assignment, and saving changes.

Coding was coordinated through GitHub, repository URL: <https://github.com/beccasbound/Project_1.git>

Assumptions

Since the assigned date is used to locate and differentiate assignments, it must be unique in the assignment book, so we assume only one assignment can be assigned on a given date. This is a drawback which could perhaps be eliminated by assigning a unique identifying number when an assignment is added to the assignment book, and including that number in the text file format.

It was assumed that input file statuses in the given text file or entered by the user would be one of the three acceptable values – assigned, completed, or late. If not, status defaults to “assigned”.

Another assumption was that the user would not not attempt to edit a completed assignment since the program does not recalculate late status based on a changed due date.

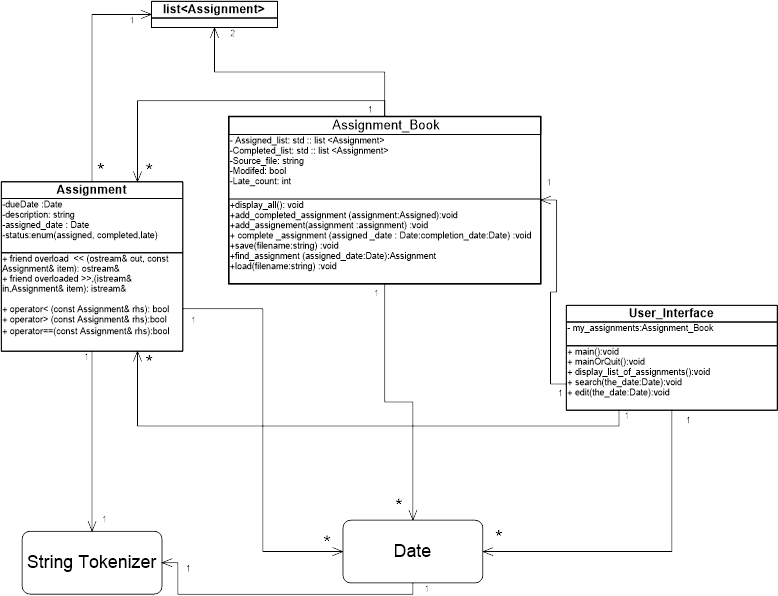
Program design does not support a user changing the completion date once entered since that data is compared to the due date immediately and not stored in a variable. Thus it was assumed that there would be no user error in entering the completion date.

Design/Implementation

The program is built on three main classes: Assignment, Assignment\_Book, and User\_Interface.

The Assignment class contains all attributes and methods of a single assignment, including due date, assigned date, description and status, along with overloaded comparison, insertion, and extraction operators. Assignments are read in using the overloaded extraction operator, then processed according to the user’s wishes via the user interface main menu. New assignments can be added through that menu as well. The user can save changes back to the text file originally read from, and the program warns a user if they have made changes and attempt to quit without saving them.

The list class comes from the standard library, and Date and String Tokenizer are code posted on Blackboard.



Performance

Big O notation

Assignment.cpp

1. Assignment (default constructor) – O(1)
2. Assignment (copy constructor) - O(1)
3. Overloaded output operator – O(1)
4. Overloaded input operator – O(n)
5. Overloaded less than operator - O(1)
6. Overloaded more than operator – O(1)
7. Overloaded assignment operator – O(1)

Assignment\_Book.cpp

1. Load function – O(n)
2. Display\_all function – O(n)
3. Find\_assignment function – O(n)
4. Add\_assignment function – O(n)
5. Add\_ completed\_assignment function – O(n)
6. Complete\_assignment function – O(n)

String\_Tokenizer.h

1. Find\_next – O(n)

Date.h

1. Constructors – O(1)
2. parseDate – O(n)
3. toString – O(1)
4. is\_leap\_year – O(1)
5. days\_of\_month – O(1)
6. days\_of\_year - O(1)
7. subtract\_days- O (n)
8. add\_days – O(n)

User\_Interface.h

Efficiency of functions in User\_Interface class depend on Assignment\_Book class’s functions which have been called which means it will have the same efficiency as Assignment\_Book functions.

Error Checking

Most error checking was done through the standard library exception class. In addition, dates were checked for validity by using the isvalid function of the Date class.

References

Date and StringTokenizer classes (files Date.h and StringTokenizer.h) from Blackboard (author: Professor Kuhail)

List class from standard library

cplusplus.com – a C++ reference website

StackOverflow.com

Professor Hare’s Tegrity Lectures spring 2016 for help with operator overloading, iterators, stl container functions.

Phone Directory code from BB – used as an example of how to organize the project into classes.