CSCI 1300 Introduction to Programming
Instructor: Boese
Homework #8

Recommendation System

Objectives

- Create a class with constructors/getters/setters/toString/helper methods
- Create and use test drivers
- Create and use vector objects
- Read code written by someone else and understand it

Recommendation System

If you've ever bought a book online, the bookseller's website has probably told you what other books you might like. This is handy for customers, but also very important for business.

Netflix Contest

For 3 years, online movie-rental company Netflix awarded one million dollars to the winners of the Netflix Prize (http://www.netflixprize.com/rules). The competition simply asked for an algorithm that would perform 10% better than their own algorithm. Making good predictions about people's preferences was that important to this company. It is also a very current area of research in machine learning, which is part of the area of computer science called artificial intelligence.

Approach

We are going to do a basic version that prints out the average ratings given by users for a list of books.

Rating System

Rating	Meaning
0	Haven't read it
-5	Hated it!
-3	Didn't like it
1	ok - neither hot nor cold about it
3	Liked it!
5	Really liked it!

Files

There are two files that we will need to read in: one for the list of books, and one for the list of users and their ratings of each book.
Example snippets of the data files are shown below:

books.txt

```
Douglas Adams,The Hitchhiker's Guide To The Galaxy
Richard Adams,Watership Down
Mitch Albom,The Five People You Meet in Heaven
Laurie Halse Anderson,Speak
Maya Angelou,I Know Why the Caged Bird Sings
Jay Asher,Thirteen Reasons Why
Isaac Asimov,Foundation Series
Ann Brashares,The Sisterhood of the Travelling Pants
```

Each line in the books.txt example file has the author, a comma, and then the title of the book.

ratings.txt

```
Ben
5 0 0 0 0 0 0 1 0 1 -3 5 0 0 0 5 5 0 0 0 0 5 0 0 0 0 0 0 0 0 1 3 0 1 0 -5 0 0 5 5 0 5 5 5 0 5 5 0 0 0 5 5 5 5 -5
Moose
5 5 0 0 0 0 3 0 0 1 0 5 3 0 5 0 3 3 5 0 0 0 0 0 5 0 0 0 0 0 3 5 0 0 0 0 0 5 -3 0 0 0 5 0 0 0 0 0 0 5 5 0 3 0 0
Reuven
5 -5 0 0 0 0 -3 -5 0 1 -5 5 0 1 0 1 -3 1 -5 0 0 0 0 0 0 3 0 0 0 0 -5 1 0 1 0 -5 0 3 -3 3 0 1 5 1 0 0 0 0 0 1 3 1 5 1 3
Cust1
3 3 5 0 0 0 3 0 0 3 0 3 0 0 0 0 0 3 0 5 0 0 0 1 3 1 0 0 0 0 0 3 0 3 0 0 0 1 3 0 0 3 3 0 0 0 5 0 0 3 1 0 0 0 0
Cust2
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 3 1 0 0 0 3 0 0 0 3 0 3 3 5 0 3 0 3
Francois
3 3 5 0 0 0 3 0 0 3 0 3 0 0 0 0 0 3 0 5 0 0 0 1 3 1 0 0 0 0 0 3 0 3 0 0 0 1 3 0 0 3 3 0 0 0 5 0 0 3 1 0 0 0 0
```

Each line in the ratings.txt example file has each user associated to two lines in the file: the first line is the user's id, and the second line is the list of ratings the user has submitted.

The way to map the ratings to the correct book is that the first rating for a given user is for the first book listed in the book file, and the second rating for a user is for the second book listed in the book file, etc. For example, Reuven gave a 5 to The Hitchhiker's Guide, a -5 for Watership Down, and a 0 for The Five People You Meet in Heaven.

Your goal is to get to the following output:

```
Reading in book list from file: books.txt
55 books read in. Closing book list file.
Reading in user list from file: ratings.txt
86 users read in. Closing user file.

Books and average ratings:
3.83 The Hitchhiker's Guide To The Galaxy by Douglas Adams
1.70 Watership Down by Richard Adams
1.83 The Five People You Meet in Heaven by Mitch Albom
1.00 Speak by Laurie Halse Anderson
-0.33 I Know Why the Caged Bird Sings by Maya Angelou
-0.33 Thirteen Reasons Why by Jay Asher
2.57 Foundation Series by Isaac Asimov
0.43 The Sisterhood of the Travelling Pants by Ann Brashares
2.50 A Great and Terrible Beauty by Libba Bray
```

```
2.25 The Da Vinci Code by Dan Brown
0.90 The Princess Diaries by Meg Cabot
3.81 Ender's Game by Orson Scott Card
3.00 The Hunt for Red October by Tom Clancy
2.50 The Hunger Games by Suzanne Collins
1.58 The Great Gatsby by F. Scott Fitzgerald
1.40 Ranger's Apprentice Series by John Flanagan
2.77 Inkheart by Cornelia Funke
2.38 Neuromancer by William Gibson
1.36 Lord of the Flies by William Golding
3.50 The Princess Bride by William Goldman
0.50 Dinotopia: A Land Apart from Time by James Gurney
1.00 Far North by Will Hobbs
0.33 Practical Magic by Alice Hoffman
1.88 Brave New World by Aldous Huxley
3.00 The Summer Tree by Guy Gavriel Kay
2.62 Flowers For Algernon by Daniel Keyes
1.00 Owl in Love by Patrice Kindl
1.52 Naruto by Masashi Kishimoto
2.33 Bleach (graphic novel) by Tite Kubo
1.00 Kiss the Dust by Elizabeth Laird
1.75 To Kill a Mockingbird by Harper Lee
3.24 The Lion the Witch and the Wardrobe by C S Lewis
3.86 The Bourne Series by Robert Ludlum
0.24 Life of Pi by Yann Martel
0.00 Breathless by Lurlene McDaniel
0.47 Twilight Series by Stephenie Meyer
4.00 Sabriel by Garth Nix
2.83 Nineteen Eighty-Four (1984) by George Orwell
2.73 Eragon by Christopher Paolini
3.58 Hatchet by Gary Paulsen
3.80 My Sister's Keeper by Jodi Picoult
2.76 The Golden Compass by Philip Pullman
3.60 Harry Potter Series by J.K. Rowling
3.56 Holes by Louis Sachar
3.00 Shonen Jump Series by Shueisha
0.00 The Shadow Club by Neil Shusterman
2.06 Bone Series by Jeff Smith
3.00 Maus: A Survivor's Tale by Art Spiegelman
2.56 The Joy Luck Club by Amy Tan
3.04 The Lord of the Rings by J R R Tolkien
2.86 The Hobbit by J R R Tolkien
2.33 Shattered by Eric Walters
2.83 The War Of The Worlds by H G Wells
2.60 Dealing with Dragons by Patricia C. Wrede
1.56 The Chrysalids by John Wyndham
```

Assignment Details:

In this assignment, you will create 3 classes:

- Book
- User
- Library

The `Book` class keeps track of books, a `User` class keeps track of each user and their ratings, and the Library will keep track of the list of Book objects and a list of User objects.

The `Library` class reads in the books, and also reads in the data file of users with their userid and ratings. The `Library` class keeps track of the list of `Book` objects in a `vector`, and the list of `User` objects in a separate `vector`.
It then calculates and prints out the average rating for each book.

The UML class diagram for the `User` and Book classes are shown below.

Book
- author : string - title : string
+ Book( ) + Book(string author, string title) + getAuthor( ) : string + getTitle( ) : string + setAuthor(string) : void + setTitle(string) : void + toString( ) : string

User
- id : string - ratings : vector<int>
+ User( ) + User(string userid, string ratings) + getId( ) : string + getRatingAt(int index) : int + addRating(int rating) : void + getNumRatings( ) : int + toString( ) : String + printRatings( ) : void

This assignment can be divided up into four parts. Think of implementing each of these as mini-assignments that build on each other (in other words, start early!).

Part 1 – Create the Book class, and test with the BookTestDriver.cpp
Part 2 – Create the User class, and test with the UserTestDriver.cpp
Part 3 – Get Part 3 to work inside the Library class
Part 4 – Get Part 4 to work inside the Library class

Parts 3 and 4 will not work unless you have Part 1 and Part 2 done and working correctly!

PART 1

The first part is to set up you Book class. Use the UML diagram and example files/output to help you figure out what functions you need to write, and what instance variables you need inside your class. For example, the toString function should return as "Book Title by Author".

You can test it with the BookTestDriver.cpp. This program calls the various functions to ensure you have it set up correctly.

PART 2

The second part is to set up your `User` class. There is also another test driver program for testing your User class named UserTestDriver.cpp.

PART 3

Once parts 1 and 2 are working correctly, then you can move on to Part 3. This time, you will be using the Library class. An outline of the Library class is provided.

For Part 3, you will notice inside the main function some comments designating Part 3. For simplification, the filenames are hard-coded in the program (I think you have enough to do). Then it calls the Library constructor with the two filenames, which then calls other functions to read in the book list and user list. *You can test by commenting out reading in the user list until you get the book list read in correctly.* Add code into the function `readInBookList` to create a new Book instance and add it to your vector of books. Then add to `readUserFile` to create a new User instance and add it to your vector of users.

PART 4

Part 4 requires Parts 1, 2, and 3 to be working properly.

For this last part, you will implement the code to print out the average rating of each book based on all the users' ratings. *Be sure to NOT include a user's rating if it is listed as a zero, as zero means that they have not read the book!*

Assignment Requirements

- The name of the files must be called **Book.cpp**, **User.cpp,** and **Library.cpp** (do not submit the test drivers).
- Comments at the top of EACH of your class files
  - Your name
  - Date
  - Homework #
  - Brief description of the class (one or two lines max)
- Example files are posted on the website for you to download. TEST WITH ADDITIONAL DIFFERENT FILES!
- You must name your classes exactly as specified.
- Your attribute names and function names must match exactly to those specified in the class diagram. You must have the same parameters as specified according to the UML class diagram. Your return values must be exactly as specified.
- The output must match exactly to the examples provided (given appropriate inputs).
- *We will run your program with different input files. So test by changing and using different example files!*
- Program must be written in C++ and submitted in Moodle.
- Zip the Book.cpp, User.cpp, and Library.cpp files and submit to Moodle as ***Firstname_Lastname_HW8.zip***.