HOMEWORK - SPRING 2022

HOMEWORK 1 - due Tuesday, February 8th no later than 7:00pm

REMINDERS:

- **Be sure your code follows the coding style for CSE214.**
- **Make sure you read the warnings about academic dishonesty. *Remember, all work you submit for homework assignments MUST be entirely your own work. Also, group efforts are not allowed.***
- **Login to your grading account and click "Submit Assignment" to upload and submit your assignment.**
- **You are not allowed to use ArrayList, Vector or any other Java API Data Structure classes to implement this assignment except where noted.**
- **You may use Scanner, InputStreamReader, or any other class that you wish for keyboard input.**

College is getting to be really expensive, and it's been getting harder and harder to find the money. Luckily, after getting bored with being rejected by Harvard, your friend Jack Ma has come up with a scheme that will make you both a bit of extra cash on the side. He has realized that you can photocopy obscure textbooks where there is no PDF available, put the pages in a binder, and rent them out for cheaper than Amazon. He has started doing this, but his service

became so popular that it was impossible to keep track of all the books he was doing. In order to not have to show up to work, he wants to be able to manipulate the text books on the shelves from his computer, and then have his workers at the warehouse do the physical labor. He has brought you to make him an electronic management system for his photocopied textbook warehouse. In this warehouse, he will have some number of shelves (he is starting with three, labeled A, B, and C), on each of which he keeps up to 20 photocopied textbooks (any more, and the shelf will break). Each textbook has a title (string), an author (string), a borrower name (string), and a condition score, ranking form 1-5 (int). Due to his perfectionism, Jack will keep his textbooks in identical boxes, left aligned on the shelf, so that they are aesthetically pleasing. Even when a textbook itself is loaned out, the box will remain on the self as a place-holder, and it will contain the vital information about the book (title, author, borrower, and condition). He wants to be able to add a new book to the collection (whenever he manages to snag a new title and photocopy it), loan books out, duplicate a textbook by photocopying it (this will use the clone() method, as we want to be able to loan the textbook copies out independently) or even a shelf of his most popular titles, swap textbook orders on the shelves (for aesthetic and organizational balance), check if two shelves are identical in content (using the equals() method), and remove sufficiently mutilated books from his system. You will write a program that simulates a warehouse with three shelves from the command line. It must follow the specifications below, and the user interface should match the interface shown below as closely as possible.

NOTE: All exceptions explicitly thrown in Required Classes except for IllegalArgumentException are custom exceptions that need to be made by you.

UML

The UML Diagram for all the classes specified below is as follows: (will be posted shortly)

Required Classes

Book
Write a fully-documented class named Book which contains the title of the textbook, (String), the author of the textbook, the name of the borrower (string), and the condition of the book (int). You should provide getter and setter methods for all member variables. In addition, you should provide a constructor as detailed below, though you may create a custom constructor which takes any arguments you see fit. You should also provide clone() and equals() methods for the Book class (note: a book's clone should NOT clone the borrower field. This should be empty in the clone; Likewise, when checking for equality, skip the borrower field). Lastly, you should provide a toString() method that returns a printable representation of the book and its data members (title, author, condition, and borrower).

- private String title
- private String author
- private String borrower
- private int condition
- public Book(String title, String author, int condition)

- Default constructor.
- Note: borrower should be set to empty string, as the book isn't loaned out yet
- Postconditions:
  - This object has been initialized to a Book object with the required properties.

- public boolean equals(Object obj)
  - Compare this Book to another object for equality.
  - Returns a value of true if the obj refers to a Book object with the same properties as this Book. Otherwise, returns a value of false. Note: you should skip checking the borrower field.

Bookshelf

Write a fully-documented class named Bookshelf which implements an ArrayList-like data structure with maximum size limited by the final int CAPACITY. The Bookshelf should contain a list of all the books on a shelf. It may not contain "holes" in the array (there should not be a non-null array element with a higher index than any null element in the array). The only exception to this is if you choose to always maintain array index 0 as null and have the array of size CAPACITY+1 so that the first book is stored internally as index 1 (otherwise, when you print to the user, you will have to add one to the book index, and subtract 1 from user input to get true index).

- private Book [] books
- private int count
- final int CAPACITY
  - this should be set to 20 or 21 depending on your implementation choice
- public Bookshelf()
  - Default constructor which initializes this object to an empty Bookshelf.
  - Postconditions:
    - The array books has been initialized, and count has been set to 0
- public int numBooks()
  - Returns the total number of books on the shelf.
  - This method should run in O(1) time.
- public Book getBook(int index)
  - Gets the reference to the Book at the given index
  - Throws an ArrayIndexOutOfBoundsException if the index is invalid
  - The list should be unchanged
- public Book removeBook(int index)
  - Gets the reference to the Book at the given index
  - Throws an ArrayIndexOutOfBoundsException if the index is invalid
  - Throws an EmptyShelfException(you must write this, an empty class extending Exception is the best way to do this) if there is no book on the shelf.
  - Removes the given book and moves all books to the right of it leftwards by one index
- public void addBook(int index, Book book)
  - Adds a book at the given index, moving all books at or after the given index one index to the right.
  - Throws an IllegalArgumentException if the index is too high and would create a hole in the array
  - Throws a FullShelfException if more than 20 books are added to the shelf (you must write this, just make an empty class that extends Exception) if the array is full.
- public void swapBooks(int index1, int index2)
  - If the indices are valid, the two books are swapped.

- Throws an ArrayIndexOutOfBoundsException if either index is invalid
- The list should be unchanged if either index was invalid

- public Bookshelf clone()
  - Creates a deep copy of this Bookshelf object (all the books are copied individually (note: the borrower field should be empty in the clone!) and placed on a new Bookshelf in the same order)
  - If the copy is modified, this object should remain unmodified.
- public boolean equals(Object o)
  - Checks if this Bookshelf is equal to another object (equal books in the same order)
- public String toString()
  - You must write a toString() method. You can use this method to help you with printing the Bookshelf for the output.

RipoffRental

Write a fully-documented class named RipoffRental which creates three instances of the Bookshelf object and provides an interface for a user to manipulate the list. Please see the UI required functions for the required functionality

- public static void main(String[] args)
  - The main method runs a menu driven application which first creates three empty Bookshelves and then prompts the user for a menu command selecting the operation. The required information is then requested from the user based on the selected operation. You can find the list of menu options in the UI required functions section.
- private static Bookshelf shelfA
- private static Bookshelf shelfB
- private static Bookshelf shelfC

You may choose to instead do the following:

- private static Bookshelf[] shelves = new Bookshelf[3];

Note: you will still ask the user to input 'a', 'b', or 'c'

Note on Exceptions: all exceptions should be handled gracefully - they should be caught in the main, and the user should be notified by a nice printout (ie: "The shelf is full, please consider visiting IKEA to get another shelf" or "There is no book at that index."). Your messages will not be graded for creativity, but they should clearly indicate what the problem is (bad index, full list, negative number, etc.). The program should continue to run normally after an exception is encountered. We will not be checking Input Mismatch cases.

Pretty Printing: Here is a tutorial on how to print tables neatly using printf in java. It is highly encouraged that you print the output neatly, as it makes grading much easier. https://docs.oracle.com/javase/tutorial/java/data/numberformat.html

---

General Recommendations

You might want to implement a toString() method for Book and Bookshelf to make debugging and printing easier. You do not have to do this, but it will help you.

You can feel free to add extra methods and variables if you need.

UI Required Functions

- A - Add Book
  - Author
  - Title
  - Condition
  - Index
- S - Swap Books
  - Index 1
  - Index 2
- L - Loan Book
  - Index
  - Recipient
  - Condition
- R - Remove Book
  - Index
- D - Duplicate Book
  - Index to copy
  - Index to insert
- C - Change Shelf
  - A, B, or C; Case Insensitive
- O - Overwrite other bookshelf with clone of the current bookshelf
  - Destination Bookshelf
- E- Check for bookshelf equality
  - 1st bookshelf
  - 2nd bookshelf
- Q - Quit

Note: please make sure that the menu is NOT case sensitive (so selecting A should be the same as selecting a).

Program Sample

```
Welcome to Jack's aMAzin' Textbook Rentals, highest price guaranteed!

Menu:
     A) Add Book
     S) Swap Books
     L) Loan Book
     R) Remove Book
     D) Duplicate Book
     C) Change Shelf
     O) Overwrite shelf with clone of current shelf
     E) Check if two shelves are equal
     P) Print current bookshelf
     Q) Quit
```

```
Please select an option: A
Please enter a title: A Brief Introduction to Brief Introductions
Please enter an author: Luke Long
Please enter condition (1-5): 4
Please enter position on shelf: 1
Book added!

//Menu not reprinted for brevity. You may reprint the menu after every instruction if you like.

Please select an option: A
Please enter a title: Introductory Computer Science
Please enter an author: TurnItOff AndBackOnAgain
Please enter condition (1-5): 5
Please enter position on shelf: 2
Book added!

Please select an option: A
Please enter a title: PRO-Crastination - A Guide To Efficiency
Please enter an author: Oscar (The Big O) Martinez
Please enter condition (1-5): 5
Please enter position on shelf: 1
Book added!

Please select an option: P

Bookshelf A:
Spot Title                                          Author                   Cond. Borrower
-------------------------------------------------------------------------------------------------------
-------
1.   PRO-Crastination - A Guide To Efficiency       Oscar Martinez           5     <none>
2.   A Brief Introduction to Brief Introductions    Luke Long                4     <none>
3.   Introductory Computer Science                  TurnItOff AndBackOnAgain 5     <none>

Please select an option: L
Please enter an index: 3
Please enter a recipient: Bill Gates
Please enter condition (1-5): 4

Introductory Computer Science has been loaned to Bill Gates.

Please select an option: S
Please enter an index: 1
Please enter another index: 2
PRO-Crastination - A Guide To Efficiency has swapped with A Brief Introduction to Brief Introductions.

Please select an option: D
Please enter a source index: 3
Please enter a destination index: 4
```

```
A new copy of Introductory Computer Science is in index 4.

Please select an option: P

Bookshelf A:
Spot Title                                              Author                   Cond. Borrower
-------------------------------------------------------------------------------------------------------
-------
1.   A Brief Introduction to Brief Introductions      Luke Long                4     <none>
2.   PRO-Crastination - A Guide To Efficiency        Oscar Martinez           5     <none>
3.   Introductory Computer Science                   TurnItOff AndBackOnAgain 4     Bill Gates
4.   Introductory Computer Science                   TurnItOff AndBackOnAgain 4     <none>

Please select an option: O
Please select shelf to overwrite with the : B
Shelf B overwritten with a copy of Shelf A.

Please select an option: C
Please select shelf to look at: B

Shelf B Selected.

Please select an option: R
Please enter an index: 2

Please select an option: P

Bookshelf B:
Spot Title                                              Author                   Cond. Borrower
-------------------------------------------------------------------------------------------------------
-------
1.   A Brief Introduction to Brief Introductions      Luke Long                4     <none>
2.   Introductory Computer Science                   TurnItOff AndBackOnAgain 4     <none> //borrower
isn't copied
3.   Introductory Computer Science                   TurnItOff AndBackOnAgain 4     <none> //duplicates
ARE allowed

Please select an option: E
Please select a shelf: a
Please select another shelf: b

These shelves are not equal.

Please select an option: C
Please select shelf to look at: A

Shelf A Selected.
```

```
Please select an option: P

Bookshelf A:
Spot Title                                              Author                   Cond. Borrower
-------------------------------------------------------------------------------------------------------
-------
1.   A Brief Introduction to Brief Introductions        Luke Long                4     <none>
2.   PRO-Crastination - A Guide To Efficiency           Oscar Martinez           5     <none>
3.   Introductory Computer Science                      TurnItOff AndBackOnAgain 4     <none>
4.   Introductory Computer Science                      TurnItOff AndBackOnAgain 4     <none>

Please select an option: R
Please enter an index: 2

Please select an option: E
Please select a shelf: a
Please select another shelf: b

These shelves are equal.

Please select an option: Q
Goodbye!
```

Course Info | **Schedule** | **Sections** | **Announcements** | **Homework** | **Exams** | **Help/FAQ** | **Grades** | **HOME**