

## **Concepts Utilized in this Project**

- Menu driven program
- A driver and a utility class
- UML diagrams
- Java fundamentals, including decision structures, loops
- Constructors, Overloaded constructors
- toString method
- Formatting output with DecimalFormat
- Random Class
- Java objects & classes

## **Overview**

Write an application that simulates adding new books to the Amazon database.

The Application will repeatedly ask for the information of a new book. Once the information is gathered, it will create an instance of a book, and display to the user. The application will then go on to the next book. Note that it will not save each book.

Information of the book includes its title, author, the number of ratings, the sum of all the ratings (the rating given to a book is 1 or 2 or 3 or 4), its price and whether the book has a hardcover or not.

When adding a new book, the user can choose to provide only the book title and its author. The rest of the information will be set to default values by the program: Price will be set to a random number between 1 and 10, the number of ratings and the sum of ratings will be set to 0 and no hardcover will be set for the book.

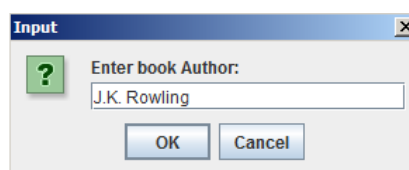
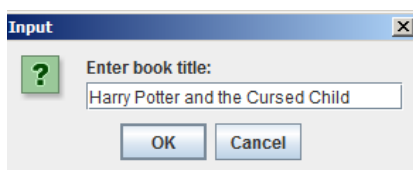
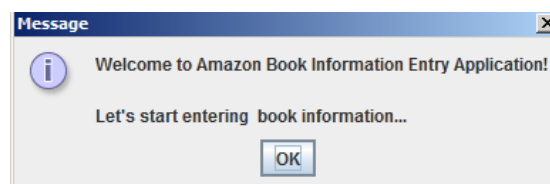
The user can also choose to enter all the book's information.

A book can have 0 or more than one rating. Rating given to a book is a number between 1 (lowest) and 4 (highest). The number of ratings and the total will be saved in the book's information.

After information is entered for the book, it will be displayed to the user. The program will also display a recommendation information for the book based on its average rating: If the average rating is between 3 and 4, the book is strongly recommended, between 2 and 3 is recommended, between 1 and 2, not recommended and if there is no rating then no information is available for recommendation.

## **User Operation**

Following is a Sample of program run:





Input

? Do you have more information for the book, ex. price, type of cover,...(y or n)?

n

OK Cancel



Input

? Enter Rating (1-4) or 0 for no rating:

4

OK Cancel

Input

? Enter more Rating (1-4) or 0 for no more rating:

3

OK Cancel



Input

? Enter more Rating (1-4) or 0 for no more rating:

9

OK Cancel



Message

i INVALID RATING!

OK

Input

? Enter Rating (1-4) or 0 for no rating:

4

OK Cancel



Input

? Enter more Rating (1-4) or 0 for no more rating:

0

OK Cancel

Message

i Following book is added to the Amazon Database:  
TITLE: Harry Potter and The Cursed Child  
AUTHOR: J.K. Rowling  
TOTAL # OF RATINGS: 3  
AVERAGE RATING: 3.67  
PRICE: 7.0  
HARDCOVER: no  
RECOMMENDATION BASED ON BOOK RATINGS: Strongly Recommended

OK



Input

? Do you want to enter another book information(y or n)?

y

OK Cancel

Input

? Enter book title:

Moonglow

OK Cancel



Input

? Enter book Author:

Michael Chabon

OK Cancel

Input

? Do you have more information for the book, ex. price, type of cover,...(y or n)?

y

OK Cancel



Input

? Enter book price:

17.49

OK Cancel

Input

? Is this Hardcover (y or n)?

y

OK Cancel



Input

? Enter Rating (1-4) or 0 for no rating:

0

OK Cancel

Message

i Following book is added to the Amazon Database:  
TITLE: Moonglow  
AUTHOR: Michael Chabon  
TOTAL # OF RATINGS: 0  
AVERAGE RATING: 0  
PRICE: 17.49  
HARDCOVER: yes  
RECOMMENDATION BASED ON BOOK RATINGS: No Rating Information Is Available

OK

## Specifications & Requirements

Design and implement a Java application to do the following:

### **Data Element Class – Book**

- Create an instance of the Book class with the given information (fields) in the overview section.
- Has a **constructor** that takes book's title and author and creates a book instance with the provided information. The number of ratings and total rating will be set to zero, price is set to random number between 1 and 10 and the book has no hardcover.
- Has another **constructor** that takes the information for the title, author, price and whether the book has hardcover or not and creates a book instance with the given information. The number of ratings and total rating will be set to zero.
- Has a method **addRating** that takes the rating for the book and adds it to the total rating as well as incrementing the number of ratings for this book.
- Has a method called **findAvgRating** that returns the average rating for this book or 0 if there is no rating for the book.
- Has a method called **bookRecommendation** that returns a string based on the average rating of the book. The book is "strongly recommended" for average rating between 3 and 4, "Recommended" for average rating between 2 and 3(exclusive) , "Not Recommended" for average rating between 1 and 2 (exclusive), and if there is no rating return "No Information Is Available For Recommendation".
- Has a **toString** method that returns the string representation of a Book object: title, author, number of ratings, average rating, price and book recommendation.

### **Driver Class - Amazon**

- This is the driver class for Book that contains a main method.
- Create a method **getInput()** that returns a Book object. This method allows the user to enter book information, uses the information to construct a Book object and returns it.
- This class contains a main method which continues asking for information about another book. It then prints each book's information, depending on the Book class' toString() method. Refer to the program sample run for more clarification.
- Add any necessary methods to modularize your code.

## Deliverables / Submissions:

Week 1: Submit your design to the assignment design folder on Blackboard one week after the project is assigned. Your design should have pseudo-code describing the actions your application will take.

Week 2: Submit two compressed files to the assignment's folder on Blackboard containing the following:

- LastNameFirstName\_Assignment3.zip, a compressed file in the zip format, with the following:
  - doc (Javadoc directory)
    - File1.html (example)
    - File2.html (example)
  - src (source code directory)
    - File1.java (example)
    - File2.java (example)
- LastNameFirstName\_Assignment3\_Moss.zip, a compressed file containing one or more Java files:
  - File1.java (example)
  - File2.java (example)
  - This folder should contain Java source files only

Be sure to review the below project rubric to understand project expectations.

## Grading Rubric CMSC 203 Assignment #2

### Overview:

There are two parts to the rubric. First, the project is graded on whether it compiles, whether it runs without errors, and whether it satisfies the specifications. These points add up to 100. Second, the score is decremented if various additional requirements are not met, e.g., no Javadoc, uses constructs that are not allowed, etc.

### PROGRAMMING

Compiles	40 pts _____
Accuracy	
Passes test cases	15 pts _____
Passes private instructor tests	15 pts _____
Execution: runs without errors (either run-time or logic errors)	30 pts _____
Possible Sub-total	100 pts _____

### REQUIREMENTS (Subtracts from Programming total)

#### Documentation:

Javadoc is not provided	- 8 pts _____
Documentation within source code is missing or incorrect	- 7 pts _____
Description of what class does is missing	
Author's Name, @author, is missing	
Methods not commented properly using Javadoc @param, @return	

#### Programming Style:

Incorrect use of indentation, statements, structures	- 4 pts _____
User interface	
Not clear to user how data is to be entered; UI does not follow requirements	- 10 pts _____
Output is difficult to understand	- 4 pts _____

#### Design:

Implementation does not match final design	- 5 pts _____
Does not use classes specified (Book and Amazon classes)	- 6 pts _____
Classes do not have the functionality specified	- 5 pts _____
Does not call both class constructors as explained in overview section	-4 pts _____
No check for invalid book rating	-1 pts _____
No Loop for entering more books	-2 pts _____
No Loop for entering more than one rating for the book	-2 pts _____
Average rating is not rounded to 2 decimal places	-1 pts _____
Price is not generated randomly when the price is not provided	-1 pts _____

Possible decrements:	-60 pts _____
Possible total grade:	100 pts _____