Object-Oriented Programming Lab#6, Fall 23

Today's Topics

- Class/Object, Constructor,
- package
- Array (Reference Type)
- ArrayList

ArrayLIst:

Action	Code
Creating an ArrayList	ArrayList<T> list = new ArrayList<T>();
Adding element to arraylist	list.add(T);
Accessing an element	List.get(int index)
Size of arraylist	list.size();

Problems/Assignments - Book Store Application

Create a book store application which will help **a book store owner** to keep the record of its books and run the business. The Book store application will help the store to keep the list of available books and have the functionalities to **1)** display all available books, **2)** sell books (should be able to sell multiple copies), **3)** order new/existing books from publishers (can create 2 option for it), and **4)** Give a discount on a book and get the discounted price. Each book in the system will have 4 attributes*;* ***bookTitle, bookAuthor, price*** *and* ***numberOfCopies***.

With **sell** or **order** of existing books, number of copies attribute will decrease/increase. With order of new book, a new book entry will be added to the system. The system will display a menu on the screen for the user to choose from. Here is the menu.

Enter "1", to display the Books: Title – Author – Quantity.
Enter "2", to order new books.
Enter "3", to sell books (need to consider discount if there is any).
Enter "4" to get the discounted price an item on sale.
Enter "5" to get total sell amount.
Enter "0", to exit the system.

Here is what you need to do to implement the Book Store.

1) Create the following **Book** class under package **bookstore**. Add the following attributes, constructor and methods.

Book
String bookTitle String bookAuthor double price int numOfCopies
Book(String, String, double, int) void getDiscountedPrice(float salePercentage) void increaseQuantity(int amt) void decreaseQuantity(int amt) String toString()

a. ***double getDiscountedPrice(float salePercentage)***method will return the price user need to pay after discount. Note: Do not change the original price.
b. ***public void increaseQuantity(int amt) –*** Increase the **numOfCopies** by **amt**.
c. ***public void decreaseQuantity(int amt) -*** decrease the **numOfCopies** by **amt**. You may need to check if you have enough copies.
d. ***Public String toString()*** method will return a String in the format "Title – Author – Quantity" format. **.**

2) Create another class "**BookStore**" under package **bookstore** which should contain all the Book objects. For now you can use an **array?ArrayList** of **Book** type and assume you can **have maximum 10 different books** (each book will have multiple copies). Or if you use **ArrayList**, no capacity restriction is needed.

BookStore
String name double totalSale Book[] books/ArrayList<Book> books
void sell(String bookTitle, String author, int noOfCopies, double salePercentage) void order(String bookTitle, String author, double price, int noOfCopies) private Book findBook(String bookTitle, String author) double getDisocuntedPrice(String bookTitle, String author, float salePercentage) double getTotalSale() void display()

a. ***sell(String, String, int, double)*** method will search for the book in "***books***" array using the **bookTitle** and **bookAuthor**. For searching call the ***findBook(…)*** method. If the book is found in the list, **number of copies** of that book will decrease. If the book is not found, a message should display. Also increment the **totalSell**. Consider if the book in on sale or not.

b. ***order(String, String, double, int)*** method will order book for the book store. You have to handle both **new** book and **existing** book scenario.
   i. First search for the book in "***books***" array using the **bookTitle** and **bookAuthor** value. For searching call the ***findBook(…)*** method.
   ii. If the book is **found** in the list (which means the book already exists in the system), **number of copies will increase**.
   iii. If the book is **not found** (which means the book does not exists in the system and you need to order new book), a new book entry will be added to the "***books***" array/arraylist.
c. ***findBook(String, String)*** method will search for the book in "***books***" array using the **bookTitle** and **bookAuthor**. If the book is found, return the item. Otherwise, return null.
d. ***double getTotalSale()*** return the **totalSale** attribute.
e. ***getDiscountedPrice(String bookTitle, String author, float salePercentage)*** method will search for the book in "***books***" array using the **bookTitle** and **bookAuthor**. For searching call the ***findBook(…)*** method. If the book is found in the list, call the ***setSalePercetage(float salePercentage)*** method.

f. ***display()*** method will display info of all books in "books" array "Title – Author – Price-Quantity" format.

3) Now create class "**BookStoreApp**" under the package **bookapp** which should contain the **main** method. In main method create an object of **BookStore** class and then provide the **menu** as mentioned before. Once the user enters his/her option, you need to read the value and take appropriate action (See below) using the **BookStore** object.

- For option 1, **display** all the books in the format above, with each one on a separate line.
- For option 2, display the information of a specific book. Ask user to enter the **bookTitle** and **bookAuthor,** call **findBooka** and then **print.**
- For option 3, the system will allow you to **order** one or more books. For this option, you need to take **bookTitle** and **bookAuthor** and ***no. of copies*** as input from user.
- For option 4, the system will allow you to **sell** one or more books. It will ask user to enter the **bookTitle, bookAuthor**, ***no. of copies*** and ***salePercentage*** to sell book.
- For option 5, view the discounted price of a book. Ask user to enter the **bookTitle** and **bookAuthor** and ***salePercetage*** and call appropriate method.
- For option 6, display the totalSale.
- For option 0, **exit** the application by breaking the loop or system exit