**Object-Oriented Programming Lab#4, Fall 2019**

**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), and **3)** order new/existing books from publishers. Each book in the system will have 4 attributes*;* ***bookTitle, bookAuthor, bookISBN*** *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 – ISBN - Quantity.

Enter “2”, to order new books.

Enter “3”, to sell books.

Enter “0”, to exit the system.

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

1. Create the following **Book** class.

|  |
| --- |
| Book |
| String bookTitle  String bookAuthor  String bookISBN  int numOfCopies |
| Book(String, String, String, int)  void display() |

* 1. ***display()*** method will display the book info in “Title – Author – ISBN – Quantity” format.

1. Create another class “**BookStore**” which should contain all the book objects. For now you ca use an **array** 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 |
| Book[] books |
| void sell(String bookTitle, int noOfCopies)  void order(String isbn, int nofOfCopies)  void display() |

* 1. ***sell(String, int)*** method will search for the book in “***books***” array using the bookTitle. If the book is found in the list, number of copies will decrease. If the book is not found, a message should display.
  2. ***order(String, int)*** method will order book for the book store. You have to handle both new book and existing book scenario.
     1. First search for the book in “***books***” array using the isbn value.
     2. If the book is found in the list (which means the book already exists in the system), number of copies will increase.
     3. 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.
  3. ***display()*** method will display info of all books in “books” array “Title – Author – ISBN – Quantity” format. Use ***Book*** class’s ***display***() method to display each book’s info.

1. Now create class “**BookStoreApp**” 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, the system will allow you to **order** one or more books. For this option, you need to take *ISBN* and *no. of copies* as input from user.
* For option 3, the system will allow you to **sell** one or more books. It will ask user to enter the *book title* and *no. of copies* to sell book.
* For option 0, **exit** the application by breaking the loop or system exit

**Phone Book Application (Assignment)**

Create a phone book application that will help a company to keep the record of phone number of its employees. For simplicity, assume you can keep at most 10 phone records. Each phone record contains the name and phone number properties. The application should allow user to *add new phone record*, *update existing record*, and *show all phone records*. The system will display a menu on the screen for the user to choose from. Here is the menu.

Enter “1”, to display the records

Enter “2”, to add new phone number.

Enter “3”, to update phone number

Enter “0”, to exit the system.

For option 1, display all the records in the format “Name: [name], Phone: [phone number]”, with each on a separate line.

For option 2, the system will allow you to add new record. For new record, you need to take name and phone number as input from user

For option 3, the system will allow you to update an existing record. Once user enters the name and the updated number, you need to find the record in the system and update the record.

For option 0, exit the application by breaking the loop or system exit.