Object Oriented Programming (OOP) Lab Task 04 E1

Task 1

1) Create a Java class named "Book" with the following attributes:

- title: String
- author: String
- price: double
- quantity: int

2) Add a constructor to the Book class that accepts all the attributes as parameters. The constructor should look like this:

`public Book(String title, String author, double price, int quantity)`

3) Add getter and setter methods for each attribute in the Book class.

4) Implement a method named `display()` in the Book class. This method should print out all the attribute values of the Book.

5) Create another Java class named "Main" which will serve as the runner or test program. Add a `main` method to this class.

Inside the Main class and main method:

1. Attempt to create an object of the Book class using the empty/default constructor. Note and resolve any errors or suggestions provided by the IDE.
2. Create two instances of the Book class using the empty constructor and set their attributes using the respective setter methods.
3. Create another two instances of the Book class using the argument constructor and provide the attribute values directly.
4. Call the `display()` method for each Book instance to print out their attribute values.
5. Instead of calling the `display()` method separately for each Book instance, use an array to store the Book instances and call the `display()` method only once for the entire array.

Task 2: Using HashMap and HashSet

In this task, you will practice using the HashMap and HashSet data structures in Java. Follow the instructions below:

1) Create a Java class named "Library".

2) Inside the Library class, declare a HashMap named "booksMap" with the key of type String and the value of type Book. This map will store books in the library, where the title of the book will serve as the key and the Book object as the value.

3) Create a method named "addBook" in the Library class that accepts a Book object as a parameter. This method should add the book to the booksMap using its title as the key.

4) Create another method named "removeBook" in the Library class that accepts a String parameter representing the title of the book to be removed. This method should remove the book with the given title from the booksMap.

5) Create a method named "displayBooks" in the Library class that iterates over the booksMap and prints out the details of each book.

6) Create a Java class named "LibraryApp" that contains the main method.

Inside the main method:

7) Create an instance of the Library class.

8) Create several instances of the Book class using different titles, authors, prices, and quantities.

9) Use the addBook method to add the Book instances to the Library's booksMap.

10) Use the removeBook method to remove a book from the Library's booksMap.

11) Call the displayBooks method to print out the details of all the books in the library.

12) Create a HashSet named "authorsSet" to store the unique authors of the books in the library.

13) Iterate over the booksMap and add the authors of each book to the authorsSet.

14) Print out the authorsSet to see the unique authors in the library.

15) Test your program by executing the LibraryApp class and verifying that the books are added, removed, and displayed correctly, and the authorsSet contains the unique authors in the library.