



National University of Computer & Emerging Sciences, Karachi  
Department of Computer Sciences  
Quiz-1 (Spring-2024)



Course No:	Course Name: Object Oriented Programming
Instructor Name :M.Minhal Raza	
Student Roll No:	Section: BSE-2A

**"If there is something, you don't know today. You will surely learn afterwards. Life is not an exam hall."**

**BEST OF LUCK!**

**Time Allowed:** 30 minutes.

**Total Marks:** 100

Instructions:

- Your Student ID must be written on the paper.

Imagine you are tasked with developing a Library System in Java. The system should be able to handle books, authors, and library members. Each book has information such as title, authors, ISBN, and availability status. Each author has a name and a unique identifier. Members of the library have a name, a unique identifier, and a list of borrowed books.

**Requirements:**

Implement a class named **Book** with the following attributes:

- **title** (String): the title of the book.
- (List<Author>) a list to who wrote the books .
- **isbn** (String): the International Standard Book Number.
- **available** (boolean): indicating whether the book is available for borrowing or not.

Implement a class named **Author** with the following attributes:

- **id** (int): a unique identifier for the author.
- **name** (String): the name of the author.

Implement a class named **Member** with the following attributes:

- **id** (int): a unique identifier for the member.
- **name** (String): the name of the member.
- **borrowedBooks** (List<Book>): a list to store the books borrowed by the member.

Implement appropriate getter and setter methods for all the attributes in each class.

Implement a constructor for each class. The constructor for the **Book** class should initialize all attributes. The constructor for the **Author** and **Member** classes should initialize only the **id** and **name** attributes.

Overload the constructors for the **Book** class. Implement a constructor that allows creating a book with only the title, author, and ISBN, assuming the book is initially available.

Implement a variable **totalBooks** in the **Book** class to keep track of the total number of books in the library. Implement a method **getTotalBooks()** to retrieve this count.

Implement a method in the **Member** class named

**borrowBook(Book book)** that adds a book to the list of borrowed books for the member. Ensure that the availability status of the book is updated accordingly.

a **returnBook** method in the Member class to simulate returning a book to the library. This method removes the book from the member's borrowed books list and updates its availability status.

a **displayBorrowedBooks** method in the Member class to display the books currently borrowed by a member.

a **displayBooks** method in the Member class to display the books

### Questions:

Demonstrate the usage of your classes by creating instances of **Book**, **Author**, and **Member**. Show how you can set and retrieve the attributes using getter and setter methods.

Create a book without specifying the availability status. Verify that the default value for availability is set correctly.

Instantiate an author and a member, and then demonstrate how the member borrows a book. Utilize the method **getTotalBooks()** to display the total number of books in the library.

```
import java.util.ArrayList;
import java.util.List;

class Book {
    private String title;
    private List<Author> authors;
    private String isbn;
    private boolean available;
    private static int totalBooks = 0;

    public Book(String title, List<Author> authors, String isbn, boolean available) {
        this.title = title;
        this.authors = authors;
        this.isbn = isbn;
        this.available = available;
        totalBooks++;
    }

    public Book(String title, List<Author> authors, String isbn) {
        this(title, authors, isbn, true);
    }

    // Getter and setter methods

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }
}
```

```

    }

    public List<Author> getAuthors() {
        return authors;
    }

    public void setAuthors(List<Author> authors) {
        this.authors = authors;
    }

    public String getIsbn() {
        return isbn;
    }

    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }

    public boolean isAvailable() {
        return available;
    }

    public void setAvailable(boolean available) {
        this.available = available;
    }

    public static int getTotalBooks() {
        return totalBooks;
    }
}

class Author {
    private int id;
    private String name;

    public Author(int id, String name) {
        this.id = id;
        this.name = name;
    }

    // Getter and setter methods

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

class Member {

```

```

private int id;
private String name;
private List<Book> borrowedBooks = new ArrayList<>();

public Member(int id, String name) {
    this.id = id;
    this.name = name;
}

// Getter and setter methods

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public List<Book> getBorrowedBooks() {
    return borrowedBooks;
}

// Other methods

public void borrowBook(Book book) {
    if (book.isAvailable()) {
        borrowedBooks.add(book);
        book.setAvailable(false);
        System.out.println(name + " borrowed " + book.getTitle());
    } else {
        System.out.println("Sorry, " + book.getTitle() + " is not available for borrowing.");
    }
}

public void returnBook(Book book) {
    borrowedBooks.remove(book);
    book.setAvailable(true);
    System.out.println(name + " returned " + book.getTitle());
}

public void displayBorrowedBooks() {
    System.out.println(name + "'s borrowed books:");
    for (Book book : borrowedBooks) {
        System.out.println("- " + book.getTitle());
    }
}

public void displayBooks(List<Book> books) {
    System.out.println("Books in the library:");
    for (Book book : books) {
        System.out.println("- " + book.getTitle());
    }
}

```

```

    }
}

public class LibrarySystemDemo {
    public static void main(String[] args) {
        // Create an author
        Author author = new Author(1, "John Doe");

        // Create a book
        List<Author> authors = new ArrayList<>();
        authors.add(author);
        Book book = new Book("Introduction to Java", authors, "1234567890");

        // Create a member
        Member member = new Member(1, "Alice");

        // Demonstrate getter and setter methods
        System.out.println("Book title: " + book.getTitle());
        book.setTitle("Java Programming");
        System.out.println("Updated book title: " + book.getTitle());

        // Demonstrate creating a book without specifying availability status
        Book bookWithoutAvailability = new Book("Advanced Java", authors, "0987654321");
        System.out.println("Availability of " + bookWithoutAvailability.getTitle() + ": " +
            bookWithoutAvailability.isAvailable());

        // Demonstrate borrowing a book
        member.borrowBook(book);
        member.borrowBook(bookWithoutAvailability);

        // Display total number of books in the library
        System.out.println("Total number of books in the library: " + Book.getTotalBooks());

        // Display borrowed books by the member
        member.displayBorrowedBooks();

        // Display all books in the library
        List<Book> allBooks = new ArrayList<>();
        allBooks.add(book);
        allBooks.add(bookWithoutAvailability);
        member.displayBooks(allBooks);

        // Demonstrate returning a book
        member.returnBook(book);

        // Display updated borrowed books by the member
        member.displayBorrowedBooks();
    }
}

```