Tutorial 1: Creating a Library Management System in Java

Setting Up Packages and Classes

1. **Create a Package**:
   - **Task**: Create a package named `library`.
   - **Code**:

```
java
Copy code
package library;
```

2. **Define Classes**:
   - **Task**: Create classes for `Book`, `Member`, and `Library`.
   - **Code**:

```
java
Copy code
public class Book {
    private String title;
    private String author;
    private String isbn;

    // Constructor
    public Book(String title, String author, String isbn) {
        this.title = title;
        this.author = author;
        this.isbn = isbn;
    }

    // Getters
    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public String getIsbn() { return isbn; }
}
```

Implementing Constructors and Getters

3. **Create Member Class**:
   - **Task**: Implement the `Member` class.
   - **Code**:

```
java
Copy code
public class Member {
    private String name;
    private int memberId;

    // Constructor
    public Member(String name, int memberId) {
        this.name = name;
        this.memberId = memberId;
```

```
    }

    // Getters
    public String getName() { return name; }
    public int getMemberId() { return memberId; }
}
```

Library Class:

- **Task**: Define the `Library` class that will manage books and members.
- **Code**:

```
java
Copy code
import java.util.ArrayList;

public class Library {
    private ArrayList<Book> books;
    private ArrayList<Member> members;

    public Library() {
        books = new ArrayList<>();
        members = new ArrayList<>();
    }

    // Method to add a book
    public void addBook(Book book) {
        books.add(book);
    }

    // Method to add a member
    public void addMember(Member member) {
        members.add(member);
    }

    // Method to list all books
    public void listBooks() {
        for (Book book : books) {
            System.out.println(book.getTitle() + " by " +
book.getAuthor());
        }
    }
}
```

Creating Methods for Library Management

5. **Implement Methods in Library Class**:
   - **Task**: Add methods to search for books and members.
   - **Code**:

```
java
Copy code
// Method to find a book by title
public Book findBookByTitle(String title) {
```

```
        for (Book book : books) {
            if (book.getTitle().equalsIgnoreCase(title)) {
                return book;
            }
        }
        return null; // Not found
    }
```

Main Class and Interaction

6. **Create Main Class**:
    - **Task**: Create a `Main` class to run the application.
    - **Code**:

```
java
Copy code
import library.*;

public class Main {
    public static void main(String[] args) {
        Library library = new Library();

        // Create books
        Book book1 = new Book("1984", "George Orwell",
"1234567890");
        Book book2 = new Book("To Kill a Mockingbird", "Harper
Lee", "0987654321");

        // Add books to the library
        library.addBook(book1);
        library.addBook(book2);

        // Create members
        Member member1 = new Member("Alice", 1);
        Member member2 = new Member("Bob", 2);

        // Add members to the library
        library.addMember(member1);
        library.addMember(member2);

        // List all books
        System.out.println("Books in the library:");
        library.listBooks();

        // Search for a book
        Book foundBook = library.findBookByTitle("1984");
        if (foundBook != null) {
            System.out.println("Found book: " +
foundBook.getTitle());
        } else {
            System.out.println("Book not found.");
        }
    }
}
```

Enhancing the System

7. **Add More Features**:
    - **Task**: Implement features like removing books, checking out books, and returning books.
    - **Code Suggestions**:
        - Create methods in the `Library` class for these functionalities.

User Input and Interaction

8. **Add User Input**:
    - **Task**: Use `Scanner` for user input to add books and members dynamically.
    - **Code**:

```
java
Copy code
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Library library = new Library();

        // Example of user input for adding a book
        System.out.print("Enter book title: ");
        String title = scanner.nextLine();
        // Continue for author and ISBN, then create and add the
book.
    }
}
```

Final Touches and Presentation

9. **Testing and Final Adjustments**:
    - **Task**: Test the application and make any necessary adjustments.
    - **Present the Project**: Have each group present their library management system, discussing their design choices and how they implemented the features.