H-TechServices

Assignment<1> - Fall 2024

Course Title:	Java	Course Code:	Java-01	Credit Hours:	4(3,1)
Course Instructor:	Muhammad Hasnat Rasool	Program Name:	Java Mastery		
Due Date:	**03-12-2024**	**Maximum Marks:**		**10**	

Important Instructions / Guidelines:

The submission date is Dec 03, 2024. Submit your assignment in the form of a report. It should contain a problem statement, solution (code), and output. Your pdf/docx file name should be your name.

Upload your file on G**ithub .**

Ensure your program runs without errors and follows the structure .

Learning Objectives: Java programming.

Question 1: Designing a Library Management System (Composition, Aggregation, Inheritance, Polymorphism, Interfaces)

Problem Statement:

You are tasked with designing a **Library Management System** where books are managed within a library. The system should allow users to search for books, borrow them, and return them. The main challenge is to model the relationships between entities using **composition** and **aggregation**, and implement **polymorphism** to handle different types of books and users.

Requirements:

1. **Composition and Aggregation**:
    - **Library**: The `Library` class should aggregate a collection of `Books` (composition).
    - **Book**: Each `Book` has attributes like `title`, `author`, and `ISBN`. Some books may be of special types like `Ebook` or `AudioBook`, demonstrating **inheritance**.

    - **User**: A `LibraryUser` can borrow and return books. Users should be able to borrow books for a certain period, and when the book is returned late, a fine should be applied.
2. **Polymorphism**:
    - Create an interface `Borrowable` with methods `borrow()` and `returnBook()`. Both `Book` and `User` should implement this interface to allow polymorphic behavior.
    - The `Book` class should implement `Borrowable`, and the `User` class should have a `borrowBook()` method that works polymorphically with different book types (e.g., `Ebook`, `AudioBook`).
3. **Inheritance**:
    - `Book` should be the base class for two subclasses: `Ebook` and `AudioBook`. Both `Ebook` and `AudioBook` should override methods like `getDetails()` and `getPrice()` to demonstrate polymorphism.
    - The `LibraryUser` class can be extended to create a subclass `AdminUser` with additional administrative rights to add or remove books from the library.
4. **Interfaces**:
    - Use interfaces for shared behaviors such as `Borrowable`, `Searchable` (for searching books by title/author), and `Returnable` (for returning borrowed books).

What is expected:

- **Composition and Aggregation**: The `Library` class should hold a collection of books, using aggregation. **Composition** is demonstrated when creating the relationship between `Book` and `LibraryUser` classes.
- **Polymorphism**: The ability to treat different book types (`Ebook` vs. `AudioBook`) in the same way by invoking the `getDetails()` or `borrow()` method on them, demonstrating polymorphism.
- **Inheritance**: The `Book` class is the parent class of `Ebook` and `AudioBook`, and `LibraryUser` can be extended to create `AdminUser`.

Output:

Library: My Library

Books in the Library:
1. Book Title: "The Java Programming", Author: "James Gosling", ISBN: "123456", Type: Physical
2. Book Title: "Effective Java", Author: "Joshua Bloch", ISBN: "654321", Type: Ebook
3. Book Title: "Clean Code", Author: "Robert C. Martin", ISBN: "111222", Type: AudioBook

Borrowing Book: Effective Java
User has borrowed "Effective Java".

Returning Book: Clean Code
Fine: 50

Marks Breakdown (Total: 10)

1. **Composition and Aggregation (2 marks)**:
    - **1 mark** for correctly implementing the **Library** class that aggregates a collection of **Book** objects (composition between `Library` and `Book`).
    - **1 mark** for correct usage of **aggregation** between `LibraryUser` and `Book`, where a `LibraryUser` can borrow and return books.
2. **Polymorphism (2 marks)**:
    - **1 mark** for creating the **Borrowable** interface with methods `borrow()` and `returnBook()`, and implementing them in the **Book** and **User** classes. Demonstrates polymorphism.
    - **1 mark** for correctly applying **polymorphism** by ensuring that different book types (e.g., `Ebook`, `AudioBook`) can be treated the same way using the `borrow()` or `getDetails()` methods.
3. **Inheritance (2 marks)**:
    - **1 mark** for creating the **Book** class and successfully extending it to **Ebook** and **AudioBook** subclasses, demonstrating **inheritance**.

- **1 mark** for ensuring that the **Ebook** and **AudioBook** subclasses override methods like `getDetails()` and `getPrice()` to provide different behaviors.

4. **Interfaces (2 marks)**:
    - **1 mark** for using interfaces like **Borrowable**, **Searchable**, and **Returnable** to create shared behaviors for books and users. This allows for a flexible and extendable system.
    - **1 mark** for ensuring the **User** class implements polymorphic behavior for different types of books (using interfaces for search and return functionality).
5. **Output/Functionality (2 marks)**:
    - **1 mark** for ensuring that the system produces the correct output showing a list of books with their details (e.g., title, author, ISBN, and type).
    - **1 mark** for correctly simulating a borrowing and returning process, including handling fines for late returns, and displaying the appropriate output for each case (e.g., borrowing a book, returning it, calculating fines).

Summary of Marks Allocation:

Task Aspect	Marks
Composition and Aggregation	2
Polymorphism	2
Inheritance	2
Interfaces	2
Output/Functionality	2
Total	**10**