American International University-Bangladesh

Course Title:	Advanced Programming with Java			Section:	A
Semester:	Spring 2024-25	Term:	Mid	Date:	27 April 2025
Type:	Lab – 1 (Makeup)	Duration:	1H45M	Total Marks:	10
Student Name:				Student Id:	

Part-A

A library is developing a new system to track book loans for its users. The system needs to handle different types of media available for borrowing — books, audiobooks, and e-books. Each type of media has a unique set of properties, but they all share some common attributes such as loan ID, username, loan date, and return date.

The library wants a system that can store and process loans efficiently and generate reports based on certain criteria. The library wants to generate reports like:

1. Total number of loans in a specific month.
2. List of loans by a specific user.
3. The average loan duration (from loan date to return date) for all loans.

The developers decide to create a flexible system that can handle different types of media, such as books, audiobooks, and e-books, without duplicating code. To achieve this, they plan to use a single solution that can store any type of media, track its details, and generate the necessary reports.

Step-by-Step Solution:

1. Identify the Common Structure for Loans:
   - All loan records have a loan ID, username, loan date, and return date.
   - We need a flexible structure that can store this common information for all types of media.
2. Use a Flexible Representation for Media:
   - Different media types (book, audiobook, and e-book) have different attributes, but the system must store all of them in a unified way.
   - We need to create a general structure (like a generic class) that can accommodate different types of media (like Book, Audiobook, E-book) as its type of parameter. This way, the system can store and manage any type of media efficiently.
3. Design the Loan Class:
   - The Loan class should store common attributes like loan ID, user name, loan date, and return date.
   - The class should also store the media type (book, audiobook, e-book) as a flexible attribute.
   - The media attribute in the Loan class should be of a generic type so that it can hold different types of media objects (e.g., Book, Audiobook, E-book).
4. Store and Manage Loans:
   - We need a collection (like a list or a map) to store multiple loan records.
   - This collection should allow us to easily add new loans, retrieve loans based on specific criteria (e.g., by user name or by date), and process loans efficiently.
5. Use Stream API for Reporting:
   - To generate reports, we should use the Stream API to filter, group, and process loan records. For example, we can:
     - Filter loans by date to count how many loans occurred in a specific month.
     - Filter loans by user name to list all loans made by a specific user.
     - Calculate the average loan duration by using the Stream API to compute the difference between the loan date and return date for each loan and then calculate the average.
6. Generate the Following Reports:
   - Total Number of Loans in a Specific Month:
     Using the Stream API, we can filter loans by the loan date (specifically the month) and count how many loans were made in that month.
   - List of Loans by a Specific User:
     We can filter the loans by the user name and generate a list of all loans made by that user.
   - Average Loan Duration:
     For each loan, calculate the duration between the loan date and return date. Then, compute the average loan duration for all loans.
7. Ensure Flexibility:
   - The system must allow for easy extension. For example, if a new type of media is introduced (e.g., podcasts), we should be able to add it to the system without changing the overall structure.
   - The use of generic classes ensures that the system can handle any type of media without needing separate code for each one.
8. Optimize the Solution:
   - Ensure that the collection of loans is processed efficiently.
   - Use the Stream API to handle filtering, grouping, and calculating values without resorting to manual iteration or looping.
   - Ensure that the solution is easy to maintain and extend as new requirements or media types are introduced.

Expected Output for Reports:

1. Total Loans in a Specific Month:
   The report should return the total number of loans for the given month. Example output:
   "Total loans in March 2025: 50"
2. Loans by User:
   The report should return a list of loans for a specific user. Example output:
   "Loans by John Doe: [Loan1, Loan2, Loan3]"

3. Average Loan Duration:
   The report should return the average loan duration in days. Example output:
   "Average loan duration: 14 days"