Question - 1

You are developing a **Java console application** to manage access to a **community library system**. The system handles **multiple users** and determines whether each user is **eligible to borrow books**, and if so, how many.

Each user must input:

- Their **name**
- Their **age**
- Their **membership type**
- Their **preferred section** (Children, Fiction, Research)

Based on this information, the application should:

1. Determine **eligibility**:
   - Only users aged **12 and above** can borrow from **Fiction** and **Research** sections.
   - Users aged **below 12** can only borrow from the **Children** section.
2. Determine how many **books** the user can borrow:
   - `"Standard"` membership:
     - Children: 3 books
     - Fiction/Research: 2 books
   - `"Premium"` membership:
     - Children: 5 books
     - Fiction/Research: 5 books
   - Any other input: Invalid membership
3. Deny access if section rules or membership rules are violated.

Additional Requirements

Implement this using:

- **Non-static methods**
- **Well-named variables and identifiers**
- **Appropriate use of control flow** (nested `if-else`, `switch`)
- **Scanner class** for input
- **Access modifiers** (`private`, `public`)
- A **proper `main` method** that:
  - Accepts number of users to process
  - Loops through each user and performs required actions

Task

Write a Java program that:

- Accepts multiple user entries.
- Uses **non-static methods** for:
  - Input collection
  - Eligibility checking
  - Book limit determination
  - Result display

- Uses **instance variables** where appropriate.
- Uses **Scanner** for reading user inputs.
- Uses **control flow** statements (`if-else`, `switch`) to implement the logic.
- Displays a final summary for each user:
    - Name
    - Age
    - Section chosen
    - Membership type
    - Eligibility status
    - Book borrowing limit (if eligible)
    - Reason if access is denied

Certainly. Here's the **expected output** for the provided program, assuming this input sequence during execution:

Sample Input

```
Enter number of users to process: 2

Processing User 1:
Enter your name: Alice
Enter your age: 10
Enter membership type (Standard/Premium): Premium
Enter section (Children/Fiction/Research): Fiction

Processing User 2:
Enter your name: Bob
Enter your age: 15
Enter membership type (Standard/Premium): Standard
Enter section (Children/Fiction/Research): Research
```

Expected Output

```
Processing User 1:

--- Access Summary ---
Name: Alice
Age: 10
Section: Fiction
Membership Type: Premium
Access Denied: Minimum age required for this section is 12.

Processing User 2:

--- Access Summary ---
Name: Bob
Age: 15
Section: Research
Membership Type: Standard
Access Granted. You can borrow 2 book(s).
```

Question - 2

Scenario-Based Question: Library Fee Calculation System

You are tasked with developing a **Java console application** for a **Library Fee Calculation System**. The system should allow users to:

1. **Input their membership status** (either "Regular" or "Premium").
2. **Input the number of days** they have borrowed books.
3. **Calculate the fee** based on the following conditions:
    - **Regular Members**:
        - **Fee per day**: 2 units of currency.
        - If the book is returned **late** (more than 14 days), an additional **fine** of 5 units per day will be charged.
    - **Premium Members**:
        - **Fee per day**: 1 unit of currency.
        - If the book is returned **late**, the fine is reduced to **3 units per day**.
    - If the membership type is invalid, print an error message.

The program should:

- Use **non-static variables** for user information (name, membership status, borrowed days).
- Use **non-static methods** to calculate the total fee and handle logic.
- Use a **Scanner class** to get user input.
- Use **if-else** and **switch** statements to check conditions and make decisions.

Requirements

- Implement methods for:
    - **Input collection** (name, membership, borrowed days).
    - **Fee calculation** based on membership type and late days.
    - **Displaying the result** with proper formatting.
- Implement proper **control flow statements** to handle logic for regular and premium members and for checking the validity of inputs.

Expected Input and Output

Sample Input 1

```
Enter your name: Alice
Enter your membership type (Regular/Premium): Regular
Enter the number of days the book has been borrowed: 20
```

Expected Output 1

```
--- Fee Calculation ---
Name: Alice
Membership Type: Regular
Days Borrowed: 20
Total Fee: 30.0 units of currency
```

Sample Input 2 (Invalid Membership)

```
Enter your name: Bob
Enter your membership type (Regular/Premium): Gold
Enter the number of days the book has been borrowed: 10
```

Expected Output 2

```
Invalid membership type.
```

Sample Input 3

```
Enter your name: Carol
Enter your membership type (Regular/Premium): Premium
Enter the number of days the book has been borrowed: 18
```

Expected Output 3

```
--- Fee Calculation ---
Name: Carol
Membership Type: Premium
Days Borrowed: 18
Total Fee: 18.0 units of currency
```

Sample Input 4 (No Fine)

```
Enter your name: David
Enter your membership type (Regular/Premium): Premium
Enter the number of days the book has been borrowed: 12
```

Expected Output 4

```
--- Fee Calculation ---
Name: David
Membership Type: Premium
Days Borrowed: 12
Total Fee: 12.0 units of currency
```

Explanation

1. **Regular Members**:
    - If the user is a **Regular Member** and has borrowed the book for **more than 14 days**, the system calculates the base fee (2 units per day) and adds an extra fine (5 units per day for days exceeding 14).
    - For borrowed days **less than or equal to 14**, there are no fines applied.
2. **Premium Members**:
    - If the user is a **Premium Member** and has borrowed the book for **more than 14 days**, the system calculates the base fee (1 unit per day) and adds an

extra fine (3 units per day for days exceeding 14).
  - For borrowed days **less than or equal to 14**, there are no fines applied.

3. **Invalid Membership**:

   - If the user inputs a membership type that is not recognized as "Regular" or "Premium", the program prints an error message and does not calculate the fee.
