Assignment Title: Book Library Management System

Objective:

Create a console-based Python application that simulates a small-scale book library system. The system should allow users to view available books, borrow and return books, and manage the library records through a text file.

Topics Covered:

- Python Basics (variables, data types, input/output)
- Loops and Conditionals
- Functions
- File Handling
- Exception Handling

Functional Requirements:

Your program must include the following features:

1. **Main Menu Loop**
   Display options continuously until the user chooses to exit:
   ```
   1. View Available Books
   2. Borrow a Book
   3. Return a Book
   4. Add a New Book
   5. Exit
   ```

2. **View Available Books**

   - Read from a text file (e.g., `library.txt`) and display books that are available.

   - Each book should have:
     ```
     Book ID | Title | Author | Status (Available/Borrowed)
     ```

3. **Borrow a Book**
    - Ask the user for a Book ID.
    - Check if the book is available.
    - Mark it as "Borrowed" in the file.
    - Handle errors: invalid Book ID, already borrowed, etc.
4. **Return a Book**
    - Ask the user for a Book ID.
    - Mark it as "Available" in the file.
    - Handle errors: invalid Book ID, already returned, etc.
5. **Add a New Book**
    - Ask the user for Book ID, Title, and Author.
    - Add the book to the file with status “Available”.
    - Handle duplicate Book IDs relevantly.

File Structure (library.txt) Example:

```
101|The Great Gatsby|F.Scott Fitzgerald|Available
102|1984|George Orwell|Borrowed
103|Pride and Prejudice|Jane Austen|Available
```

Technical Requirements:

- Use **functions** for modularity (`add_book()`, `borrow_book()`, etc.)
- Use **loops** to manage the menu system
- Use **conditionals** for decision making
- Implement **file handling** to read/write book data
- Use **exception handling** to manage invalid inputs and file errors
    - Invalid input
    - File not found or corrupted data

Submission Requirements:

- Python `.py` file (well-commented)
- Sample `library.txt` file with at least 10 - 15 books
- Readme or comment file explaining how to run the program

Evaluation Criteria:

Criterion	**Marks**
Functional Correctness	30
Code Organization (Functions, Loops, Conditionals)	20
File Handling Accuracy	15
Exception Handling	15
User Interface and Usability	10
Code Comments and Clarity	10
Total	**100**