Day 7 Task

Create a modular Library Management System with different modules for book management, member management, borrowing and returning books, and report generation. Each module should encapsulate related functionalities, promoting code organization and reusability. Utilize various programming concepts such as error handling, data structures (lists, sets, tuples, dictionaries), lambda functions, list comprehension, classes, objects, instance variables/methods, static variables/methods, special methods (__init__, __str__, __repr__), inheritance, polymorphism, encapsulation, modules (datetime, math, os), and exception handling.

Breakdown:

1. Book Management Module:
   a. Create a module named book_management with a class named Book to handle book-related operations.
   b. Implement methods to add new books, update book details, and remove books.
   c. Utilize a dictionary within the module to maintain the book inventory.

2. Member Management Module:
   a. Create a module named member_management with a class named LibraryMember for member-related operations.
   b. Implement methods to add new members, update member details, and remove members.
   c. Utilize a list within the module to maintain the list of library members.

3. Borrowing and Returning Module:
   a. Create a module named borrow_return to handle borrowing and returning book operations.
   b. Implement functions to check book availability, update book quantity, and handle member book transactions.

4. Report Generation Module:

a. Create a module named report_generation with functions to generate various reports.

b. Implement a function to display the current book inventory, sorting options, and member borrowing reports.

c. Use the datetime module for timestamping and the os module for directory creation.

5. Error Handling Module:

   a. Create a module named error_handling with classes or functions to handle different types of errors.

   b. Implement error handling for scenarios like invalid ISBN, negative quantity, or non-existent members.

6. Datetime Module:

   a. Utilize the datetime module within the relevant modules to record timestamps for book transactions.

7. Math Module:

   a. Use the math module within the report generation module to calculate fines for overdue books.

8. Os Module:

   a. Leverage the os module within the report generation module to create a directory structure for storing reports and logs.

By organizing the functionalities into separate modules, you ensure a clean and modular design for the Library Management System, making it easier to maintain, extend, and collaborate on different aspects of the system.