PYTHON OOP Exercise #5

Classes : Books, Students, Librarian
Tasks:
Students can borrow books, students can return books, and students can check available books.
Librarians can add books, remove books and check available books.

```
class Book:
    def __init__(self, title, author, copies):
        self.title = title
        self.author = author
        self.copies = copies


    def check_availability(self):
        return self.copies


    def borrow(self):
        if self.copies > 0:
            self.copies -= 1
            return f"Successfully borrowed '{self.title}' by {self.author}."
        else:
            return f"Sorry, '{self.title}' is currently unavailable."


    def return_book(self):
        self.copies += 1
        return f"Returned '{self.title}' by {self.author}."

    def __str__(self):
      return f"'{self.title}' by {self.author} ({self.copies} available)"


class Librarian:
    def __init__(self):
        self.books = []


    def add_book(self, title, author, copies=1):
        for book in self.books:
            if book.title == title and book.author == author:
                book.copies += copies
                return f"Added {copies} copies of '{title}' by {author}."
        new_book = Book(title, author, copies)
        self.books.append(new_book)
        return f"Added '{title}' by {author} with {copies} copies."
```

```
    def remove_book(self, title, author):
        for book in self.books:
            if book.title == title and book.author == author:
                self.books.remove(book)
                return f"Removed '{title}' by {author}."
        return f"'{title}' by {author} not found in the library."


    def list_books(self):
        for book in self.books:
            print(book)


class Student:
    def __init__(self, name):
        self.name = name


    def borrow_book(self, book):
        return book.borrow()


    def return_book(self, book):
        return book.return_book()


# Example usage
if __name__ == "__main__":
    librarian = Librarian()


    librarian.add_book("Python Crash Course", "Eric Matthes", 5)
    librarian.add_book("The Pragmatic Programmer", "Andrew Hunt and David Thomas", 3)
    librarian.add_book("Clean Code", "Robert C. Martin", 4)


    student1 = Student("Alice")
    student2 = Student("Bob")


    print("Books available in the library:")
    librarian.list_books()


    print("\nAlice borrows a book:")
    print(student1.borrow_book(librarian.books[0]))
```

```
print("\nBooks available in the library:")
librarian.list_books()


print("\nAlice returns the book:")
print(student1.return_book(librarian.books[0]))


print("\nBooks available in the library:")
librarian.list_books()


print("\nBob borrows a book:")
print(student2.borrow_book(librarian.books[0]))


print("\nBooks available in the library:")
librarian.list_books()


print("\nLibrarian removes a book:")
print(librarian.remove_book("Python Crash Course", "Eric Matthes"))


print("\nBooks available in the library:")
librarian.list_books()
```

Instruction on Solving the problem:

Step 1: Define the Classes

- Start by defining the three main classes: Book, Librarian, and Student.
- Book class represents the books in the library.
- Librarian class manages the library's books.
- Student class represents the students who can borrow and return books.

Step 2: Implement the Book Class

- In the Book class, create an __init__ method to initialize book attributes (title, author, copies).
- Implement methods to check book availability, borrow a book, and return a book.
- The __str__ method is used to display book information.

Step 3: Implement the Librarian Class

- In the Librarian class, create an __init__ method to initialize an empty list to store books.
- Implement methods to add books, remove books, and list available books in the library.

Step 4: Implement the Student Class

- In the Student class, create an __init__ method to store the student's name.
- Implement methods for borrowing and returning books.

Step 5: Example Usage

- In the example usage section, create instances of the Librarian and Student classes.
- Add books to the library using the add_book method of the Librarian class.
- Demonstrate students borrowing and returning books using the borrow_book and return_book methods of the Student class.