**Assignment 8: ES6+ Features**

**Task: Rewrite a previous assignment (e.g., Assignment 3 or 5) using ES6+ features such as arrow functions, template literals, and destructuring.**

**Let's take the library system from Assignment 3 and rewrite it using ES6+ features such as arrow functions, template literals, and destructuring.**

| **// ES6+ Library System**  **class Book {**  **constructor(title, author, ISBN) {**  **this.title = title;**  **this.author = author;**  **this.ISBN = ISBN;**  **this.isAvailable = true;**  **}**  **displayInfo() {**  **console.log(`Title: ${this.title}`);**  **console.log(`Author: ${this.author}`);**  **console.log(`ISBN: ${this.ISBN}`);**  **console.log(`Available: ${this.isAvailable ? 'Yes' : 'No'}`);**  **console.log('----------------------');**  **}**  **}**  **class Patron {**  **constructor(name, libraryCardNumber) {**  **this.name = name;**  **this.libraryCardNumber = libraryCardNumber;**  **}**  **displayInfo() {**  **console.log(`Name: ${this.name}`);**  **console.log(`Library Card Number: ${this.libraryCardNumber}`);**  **console.log('----------------------');**  **}**  **}**  **class Transaction {**  **constructor(book, patron, dueDate) {**  **this.book = book;**  **this.patron = patron;**  **this.dueDate = dueDate;**  **this.isReturned = false;**  **}**  **displayInfo() {**  **console.log('Transaction Details:');**  **console.log('----------------------');**  **this.book.displayInfo();**  **this.patron.displayInfo();**  **console.log(`Due Date: ${this.dueDate}`);**  **console.log(`Returned: ${this.isReturned ? 'Yes' : 'No'}`);**  **console.log('----------------------');**  **}**  **returnBook = () => {**  **this.isReturned = true;**  **this.book.isAvailable = true;**  **console.log(`Book "${this.book.title}" has been returned by ${this.patron.name}.`);**  **};**  **}**  **// Example usage**  **const book1 = new Book('The Hobbit', 'J.R.R. Tolkien', '9780547928227');**  **const book2 = new Book('To Kill a Mockingbird', 'Harper Lee', '0061120081');**  **const patron1 = new Patron('John Doe', 'A12345');**  **const patron2 = new Patron('Jane Smith', 'B67890');**  **const transaction1 = new Transaction(book1, patron1, '2024-02-01');**  **const transaction2 = new Transaction(book2, patron2, '2024-03-01');**  **// Display book and patron information**  **book1.displayInfo();**  **book2.displayInfo();**  **patron1.displayInfo();**  **patron2.displayInfo();**  **// Display transaction information**  **transaction1.displayInfo();**  **transaction2.displayInfo();**  **// Simulate returning a book**  **transaction1.returnBook();**  **transaction1.displayInfo(); // Display updated transaction information** |
| --- |

In this rewritten version, I have used arrow functions for class methods, template literals for string interpolation, and destructuring where applicable. Arrow functions are particularly useful for concise method definitions and maintaining the lexical context (i.e., this binding). Template literals provide a cleaner and more readable way to concatenate strings, and destructuring helps simplify the code by extracting values from objects or arrays