Object-Oriented Programming
Group Assignment #1: Library Book Management System

CMP_SC/INFO_TC 3330

Spring 2025

1 Objective

The objective of this assignment is to implement a simple **Library Book Management System** using Java. You will apply object-oriented programming principles such as encapsulation, constructors, accessor/mutator methods, and object comparison.

2 Problem Description

You will create a `Book` class that represents a book with attributes such as title, author, ISBN, and price. Then, you will implement a `Library` class that manages an array of books and provides functionalities to add, search, and display books.

3 Part 1: The `Book` Class (20 points)

Create a `Book` class with the following specifications:

- **Private attributes**:
  - `String title`
  - `String author`
  - `String ISBN`
  - `double price`
- **Constructors**:
  - A **default constructor** that initializes title, author, and ISBN to `"Unknown"` and price to `0.0`.
  - A **parameterized constructor** to set all attributes.

  - A **copy constructor** that creates a new `Book` object from another `Book`.

- **Methods**:
  - `public String getTitle()`, `getAuthor()`, `getISBN()`,
  - `public double getPrice()`
  - `public void setTitle(String title)`, `setAuthor(String author)`, `setISBN(String ISBN)`, `setPrice(double price)`
  - `public String toString()` - Returns a string representation of the book. (Must be overridden)
  - `public boolean equals(Book other)` - Returns `true` if two books have the same ISBN. (Must be overridden)

4 Part 2: The `Library` Class (20 points)

Create a `Library` class to manage an array of `Book` objects.

- **Attributes**:
  - A private array `Book[] books` with a fixed size of 5.
  - A private integer `count` to keep track of the number of books.

- **Methods**:
  - `public boolean addBook(Book book)` - Adds a book to the array (if there is space).
  - `public boolean removeBook(Book book)` - Removes a book from the array (if it exists).
  - `public Book searchByISBN(String ISBN)` - Searches for a book by ISBN and returns the `Book` object (or `null` if not found).
  - `public void displayBooks()` - Prints details of all books using `toString()`.

5 Part 3: Test Class (5 points)

Create a `LibraryApp` class with a `main()` method that:

- Creates a `Library` object.
- Adds at least **3 books** to the library. Don't add books with user inputs, I hate user inputs. Instead, add your book input through your setter methods and constructors. In other words, have you inputs hard coded, it simplifies things.

- Searches for a book by ISBN and displays the result.
- Remove a book that exists and does not exist.
- Displays all books in the library.

6 Important Notes

- Follow Java naming conventions, or you will lose points.
- Use packages or you will lose points.
- Add Javadoc to your code, or you will lose points.
- Export your project properly, or you will lose points.
- Don't want to 1 commit project, and commit messages like "*Adding Java code*" or "*Update code*", otherwise you will lose points. Commits must be small and meaningful with a commit message that is relevant to the code you pushed.
- Write your code considering edge cases. Make sure you have error controls.
- Don't ask how much points will be deducted for the notes above. There is no negotiation here. These are good practices that you must adopt and follow to have a successful career. You can try to violate one of the good practices above and see what happens :) (not recommended).
- Everyone in the group must contribute to the project. Use Git efficiently and communicate!
- If there is a group drama, you have to wait until the next group assignment to split from your group, or work alone. See syllabus for details.
- **Due date:** 2/12/2025, 11:59 PM.
- **Submission:** You must submit your GitHub repository, and your exported project through Canvas.

7 Sample Output

```
Adding books...
Book added successfully.

Searching for book with ISBN: 978-0135166307
Book found: Clean Code by Robert C. Martin (ISBN: 978-0135166307, $40.99)

All books in the library:
1. Clean Code by Robert C. Martin (ISBN: 978-0135166307, $40.99)
```

```
2. Design Patterns by Erich Gamma (ISBN: 978-0201633610, $50.99)
3. The Pragmatic Programmer by Andrew Hunt (ISBN: 978-0135957059, $45.99)

Removing book: Design Patterns by Erich Gamma (ISBN: 978-0201633610, $50.99)

All books in the library:
1. Clean Code by Robert C. Martin (ISBN: 978-0135166307, $40.99)
2. The Pragmatic Programmer by Andrew Hunt (ISBN: 978-0135957059, $45.99)

Cannot Remove book ‘‘Design Patterns by Erich Gamma (ISBN: 978-0201633610, $50.99)", book do
```