CA 3: Experiential Learning

Group Members:

| Sr. No. | PRN | Name of Student | Mail id |
| --- | --- | --- | --- |
| 1 | 22070122086 | Janmejay Pandya | janmejay.pandya.btech2022@sitpune.edu.in |
| 2 | 22070122089 | Josh Dagla | josh.dagla.btech2022@sitpune.edu.in |
| 3 | 22070122103 | Kush Deo Shukla | kushdeo.shukla.btech2022@sitpune.edu.in |

***Problem Statement****: Design a C++ program for a Teacher Library Management System that allows teachers to manage their personal library collections. The program should provide a user-friendly interface for teachers to add, view, update, and checkout books from the library. Its key features will comprise of:-*

1)**Teacher Profiles:**

* Teachers will be able to set up their profiles with unique identifiers such as Usernames and Employee IDs.

2)**Book Management:**

* The system will consist of features such as adding books to the library with information like their titles and number of copies added.
* Viewing a list of all the books stored in the library with the titles and number of copies available.
* Updating books and any necessary information about them.
* Removing books from the library.

3)**Borrowing and Returning:**

* Teachers will be able to borrow and return books with ease.
* By accessing their profile, Teachers will also be able to know necessary details like the amount of books issued with title.

4)**Searching Function:**

* Books can be searched based off of the title to check availability and amount available.

5)**User Friendly Interface:**

* The text-based user interface will be easy to access and use for teachers.

6)**Error Handling:**

* Any sort of error and issues will be solved by displaying text-based messages in the system to guide the Teachers and make the overall user experience better.

***Explanation***: *This application, which is written in C++, simulates a fundamental library management system. It provides a menu-driven interface for users to carry out different tasks related to adding, displaying, and checking out books and teachers. It specifies two classes, Book and Teacher. The functionality of the code is broken down as follows:*

**Book Class:**

* It has the characteristic *‘title’* and *‘number\_of\_copies’* to represent a book.
* These attributes are initialized by the constructor.
* Provides methods for:
* *display():* Shows the title and number of copies available of the book.
* *isAvailable():* Determines if the book is available for checkout and how many copies of the book are available for checkout.
* *checkoutBook():* If the book is available, it is checked out, reducing the number of copies that are still available.
* *getTitle():* Returns the book's title.

**Teacher Class(Inherits from book):**

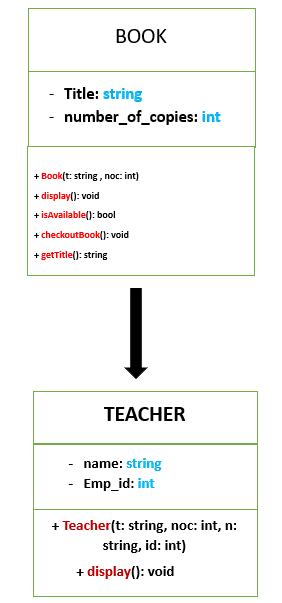
* Represents a teacher, adding its own attributes, *‘name’* and *‘Emp\_id’*, and inheriting the properties and methods from the *‘Book’* class.
* Both the properties unique to the teacher and those derived from *‘Book’* are initialized in the constructor.
* Overrides the *‘display()’* function to show the employee ID and name of the teacher in addition to the book details.

**Main Function:**

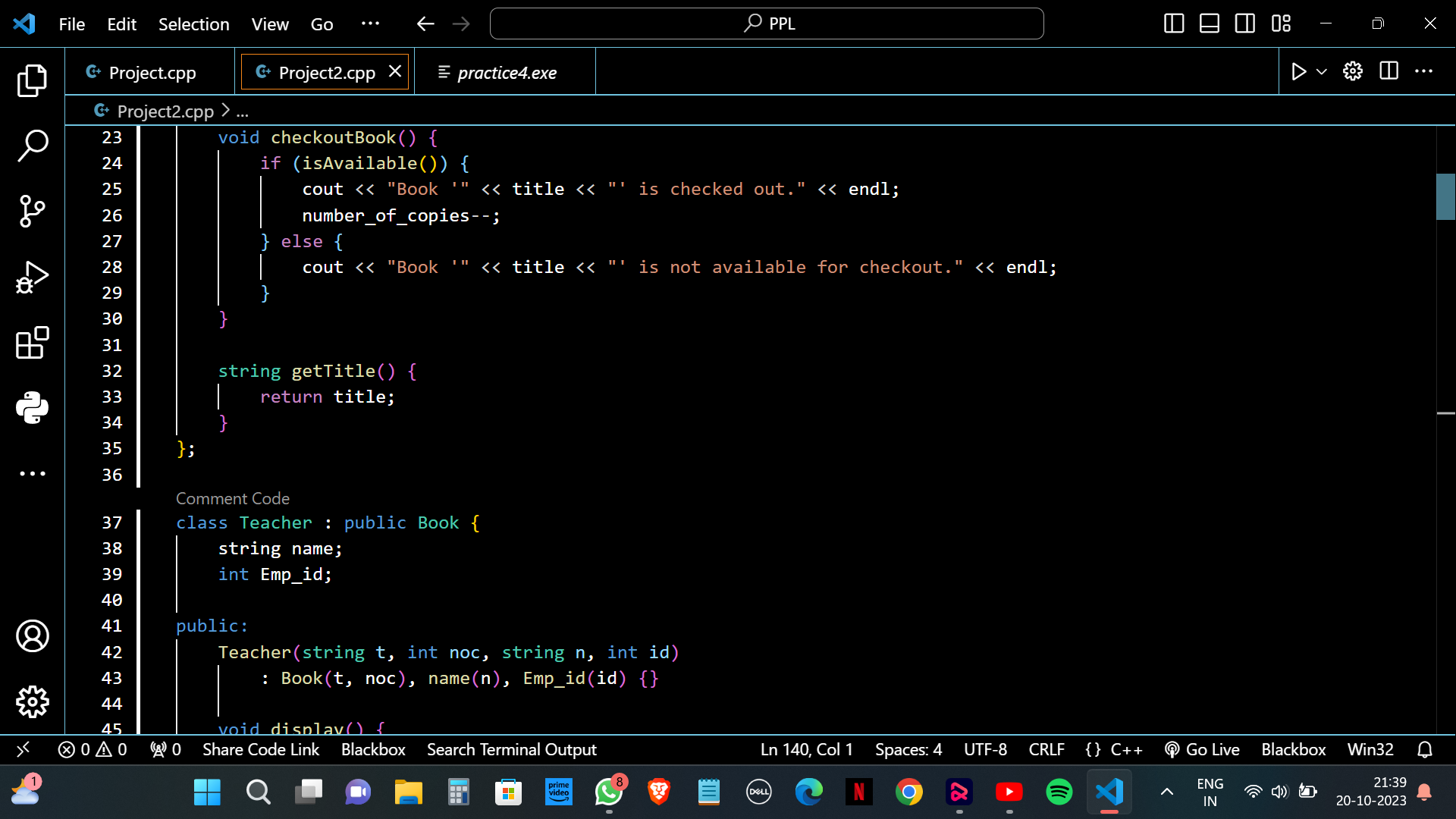
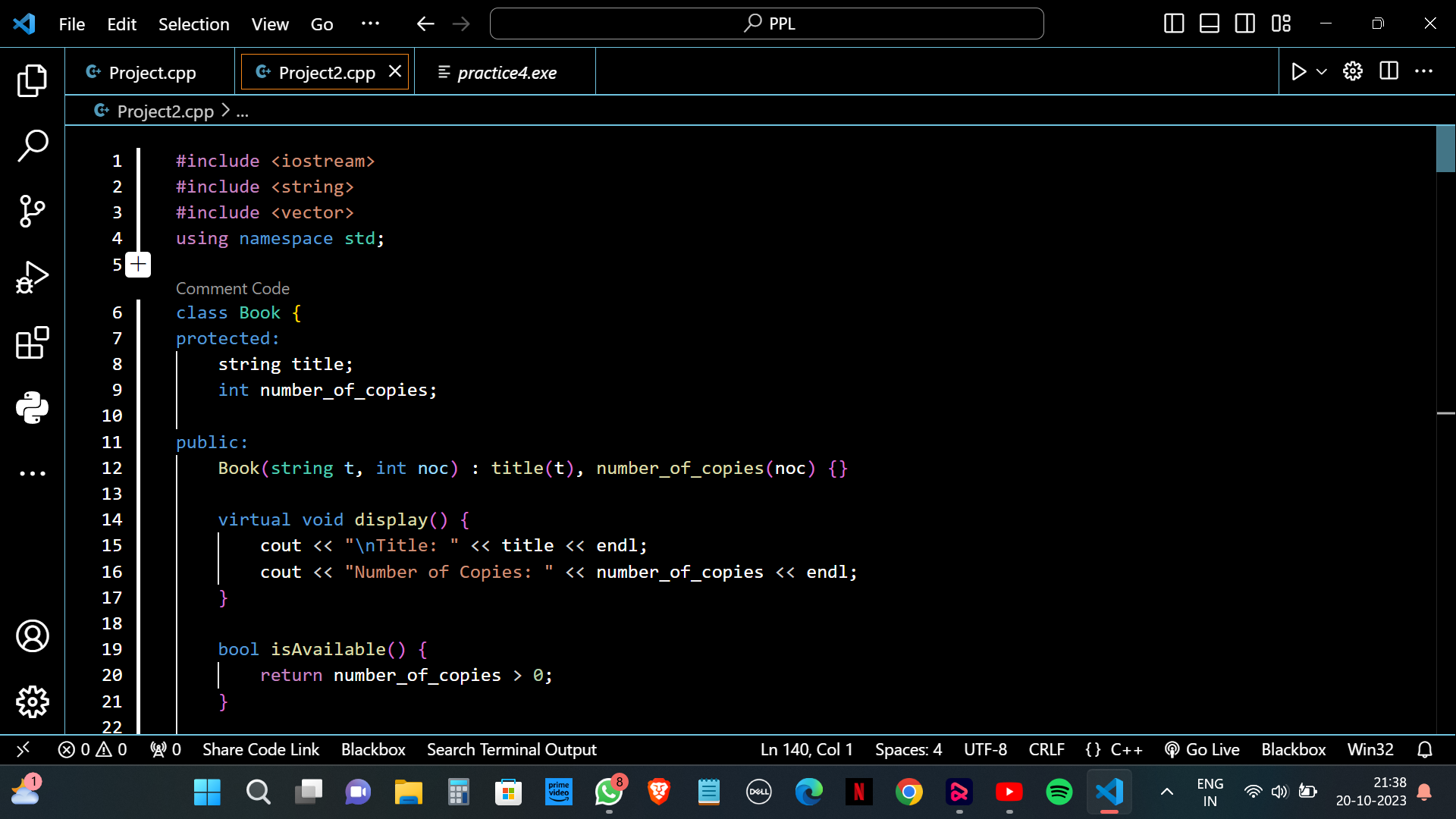
* Initializes vectors to store pointers to Book and Teacher objects: Books and Teachers.
* Uses a while loop to show the user a menu-driven interface.
* The menu choices consist of:
* Adding a fresh book, along with details such as how many copies there are.
* Adding a new Teacher and providing them with their name, employment ID, the title of the book they would like to borrow, as well as the quantity of copies.
* Showing all Books or all Teachers while listing their information.
* Requesting the title of a book to be checked out. It lowers the number of copies if it is available.
* Clearing up dynamically allocated objects (deleting those kept in vectors) and exiting the program.

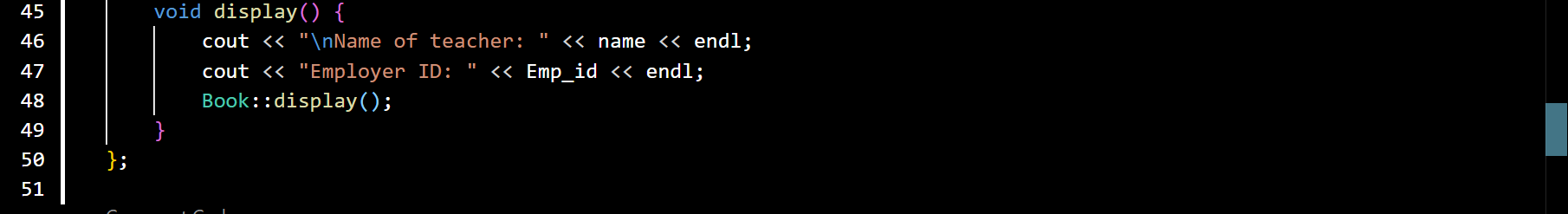
*‘Book’* and *‘Teacher’* objects are dynamically allocated memory by the code, and references to these objects are stored in the vectors. Users can communicate with the system using a text-based menu, and it offers minimal capability for managing a collection of books and teachers.

*Class Diagram:*

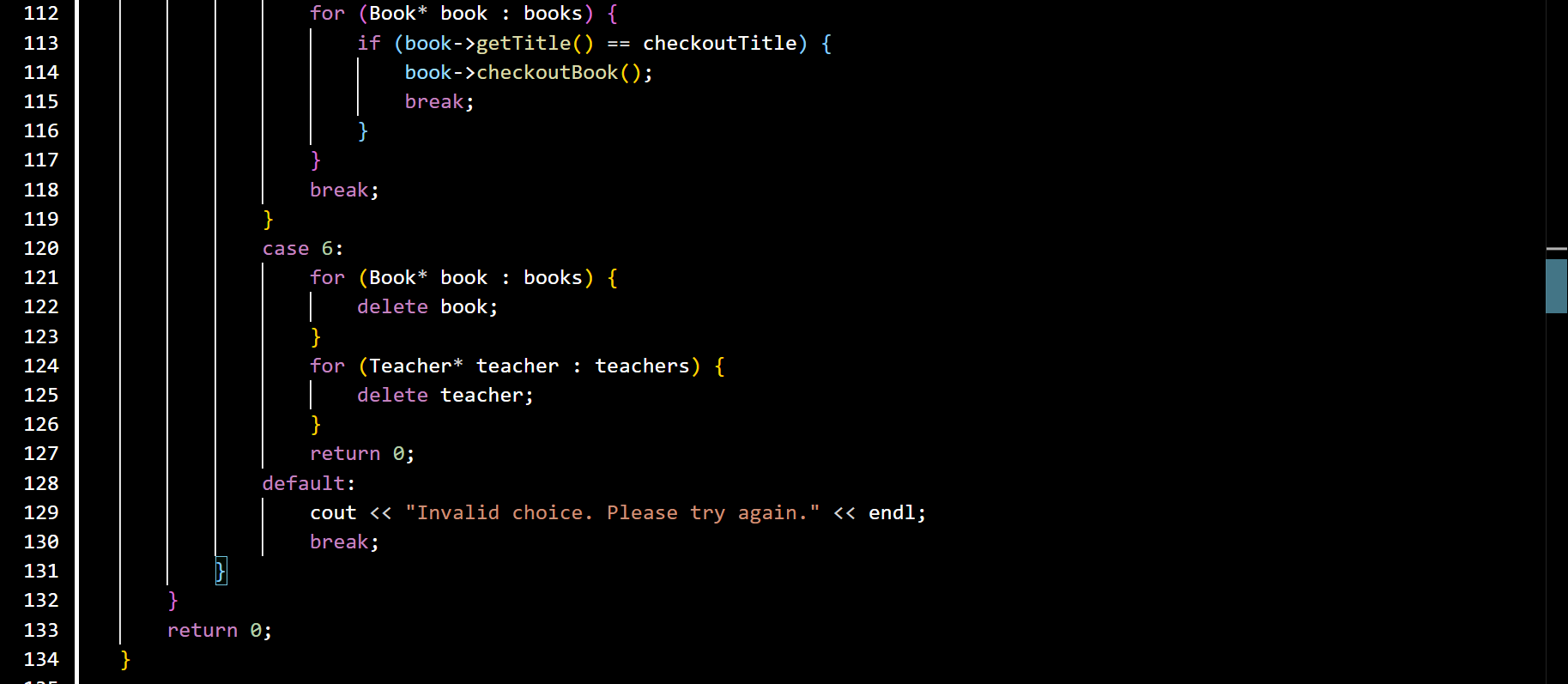
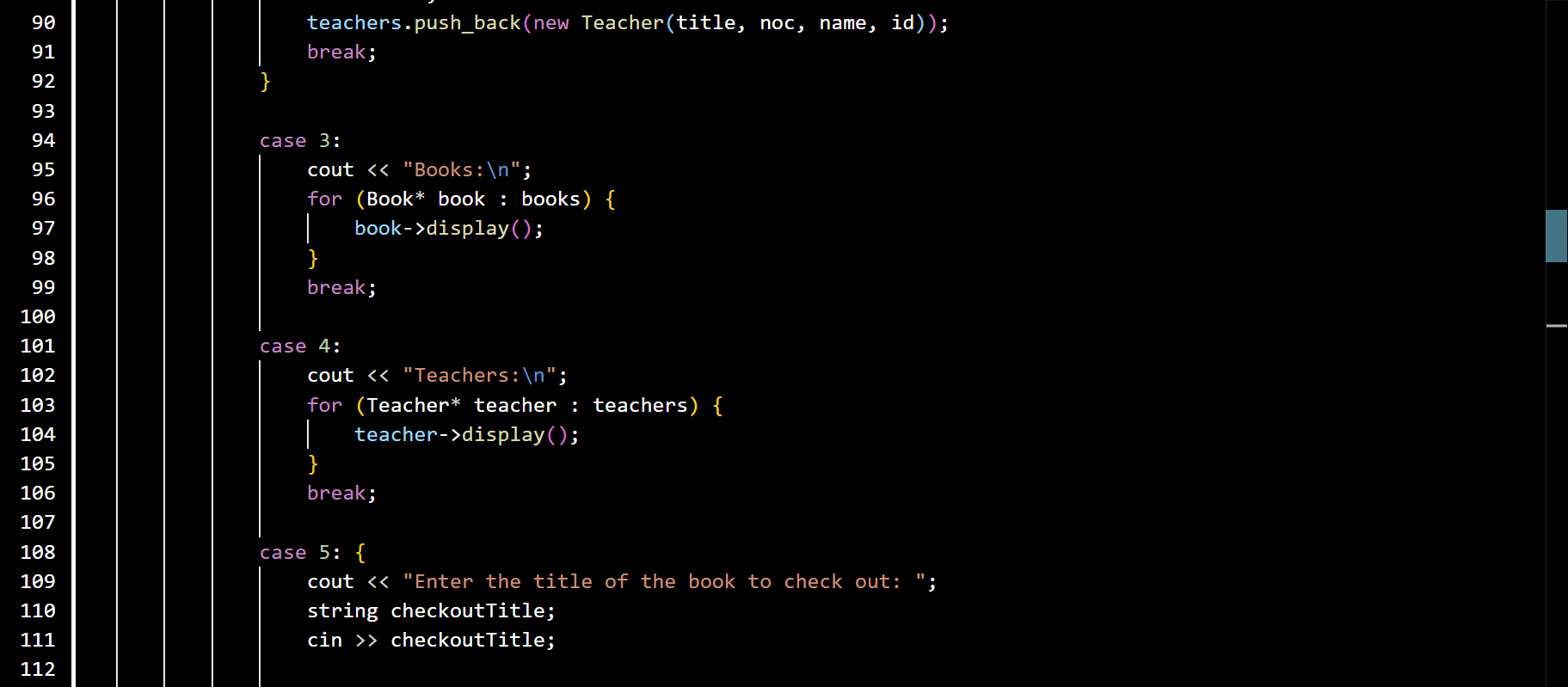
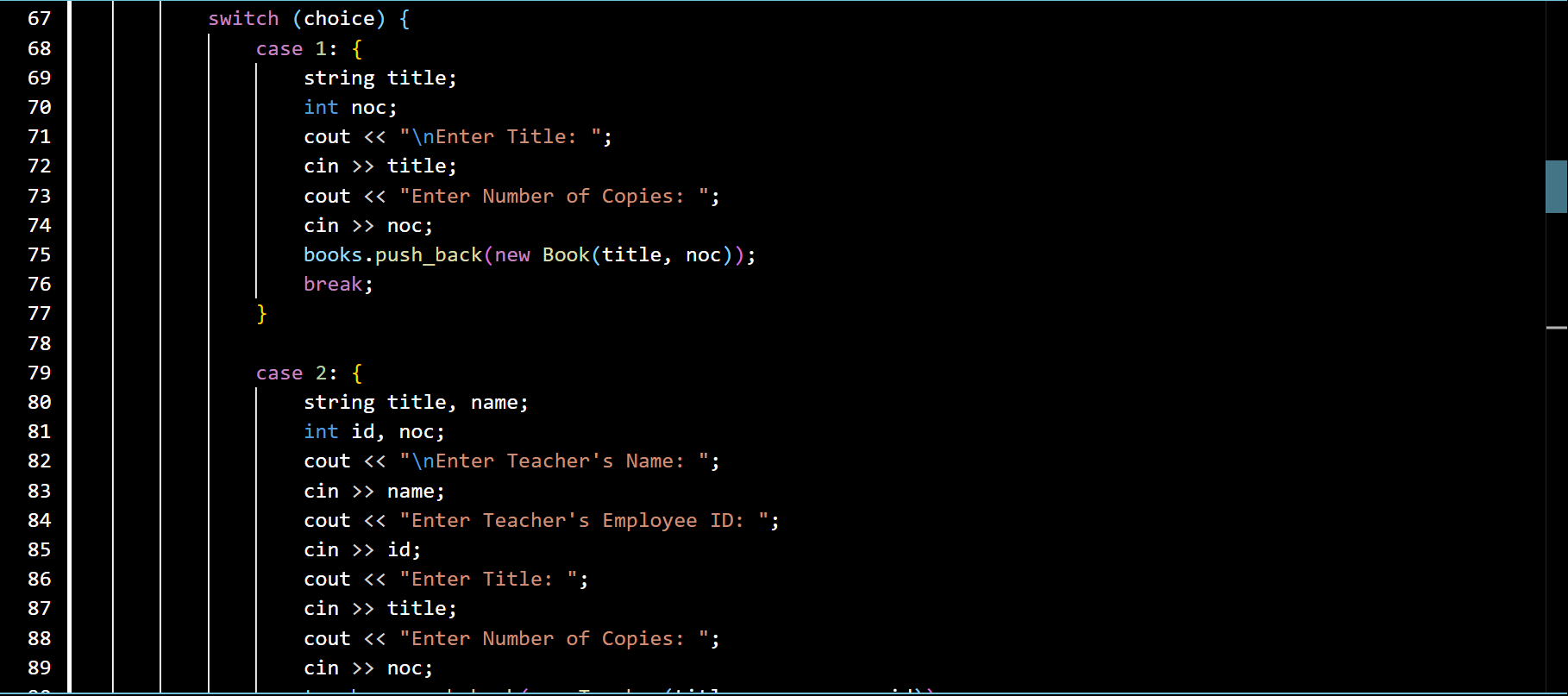
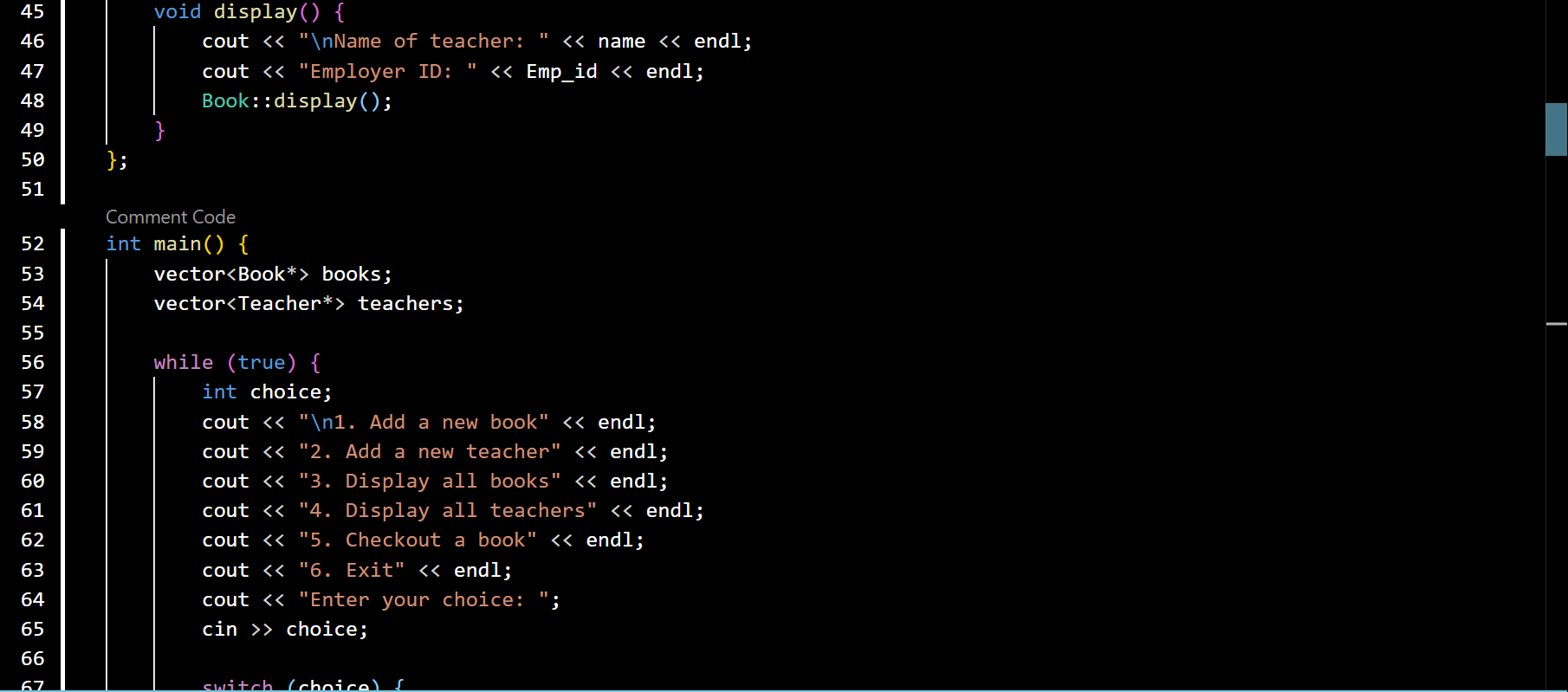
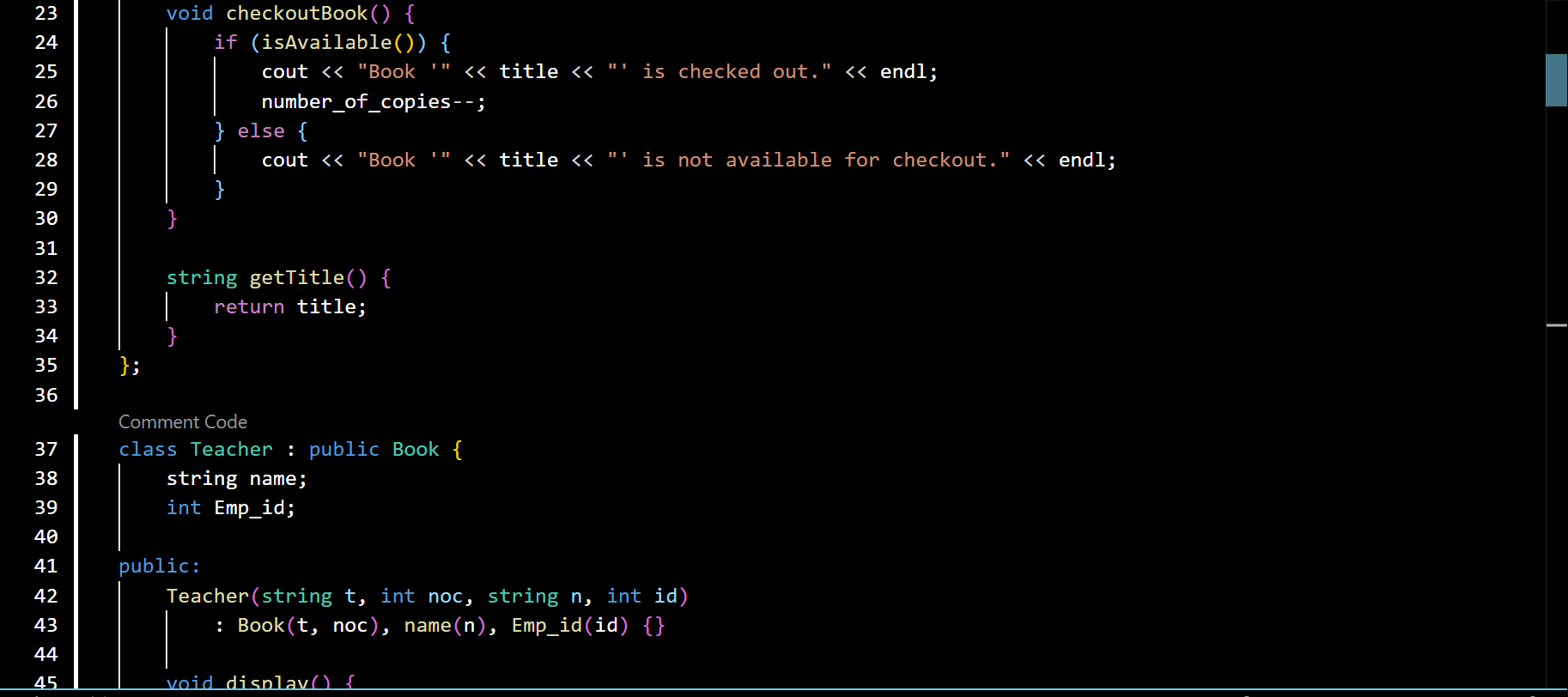
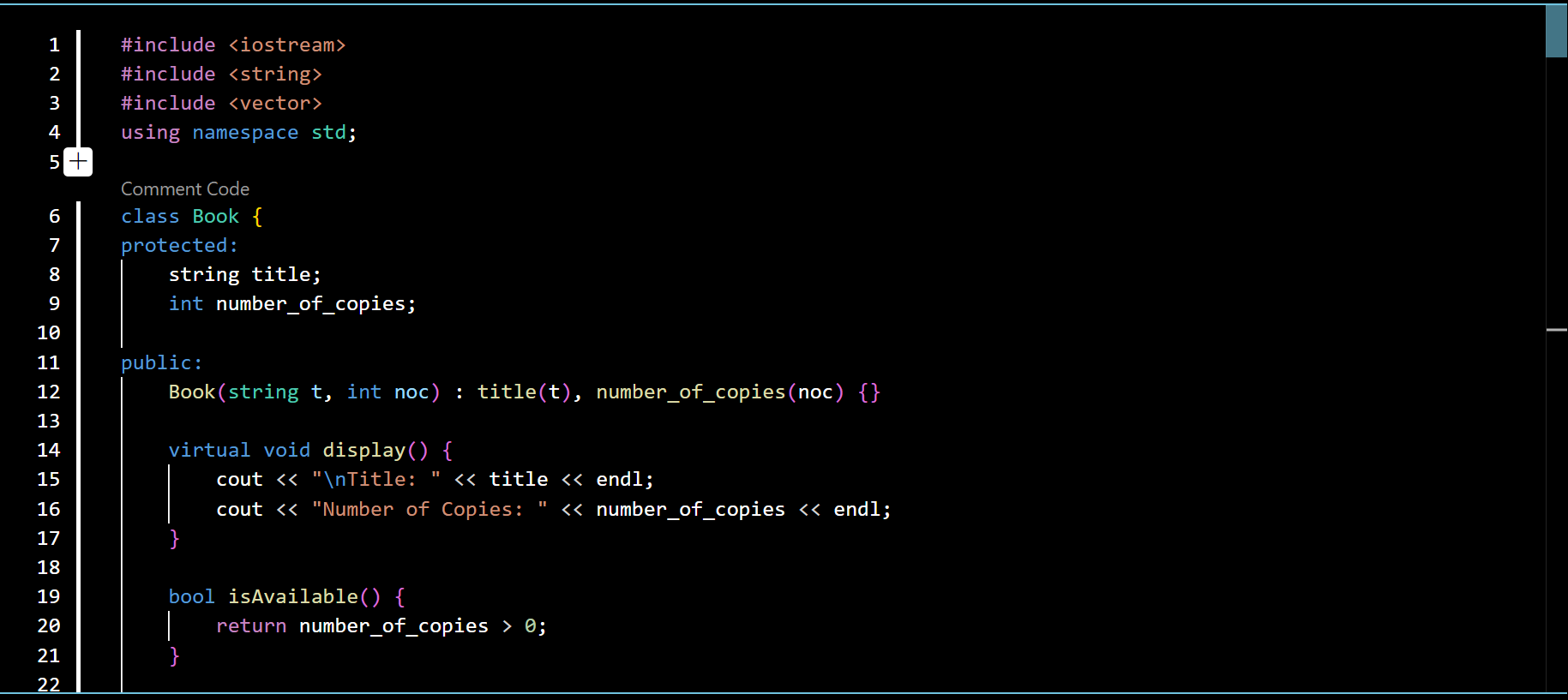


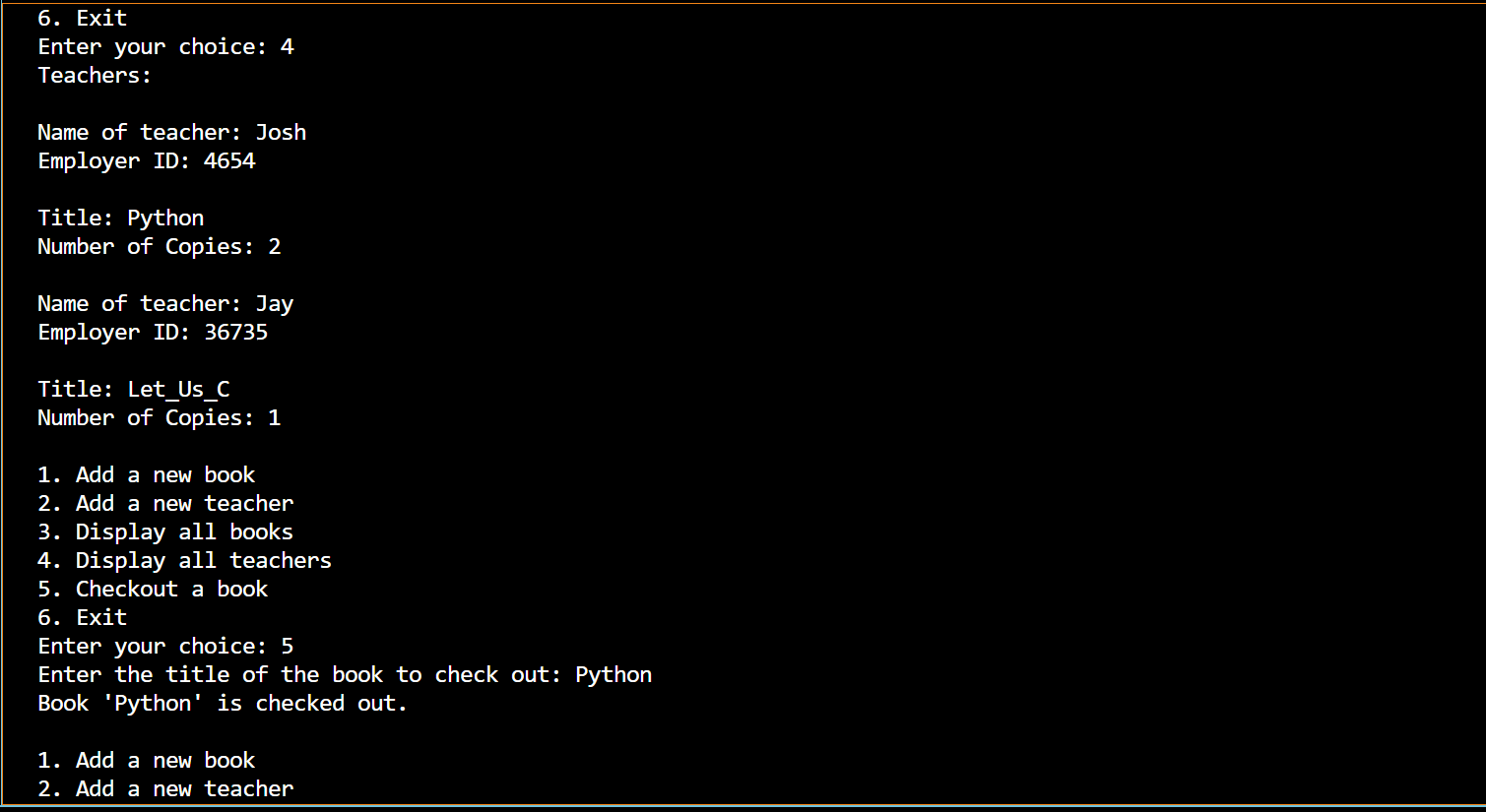
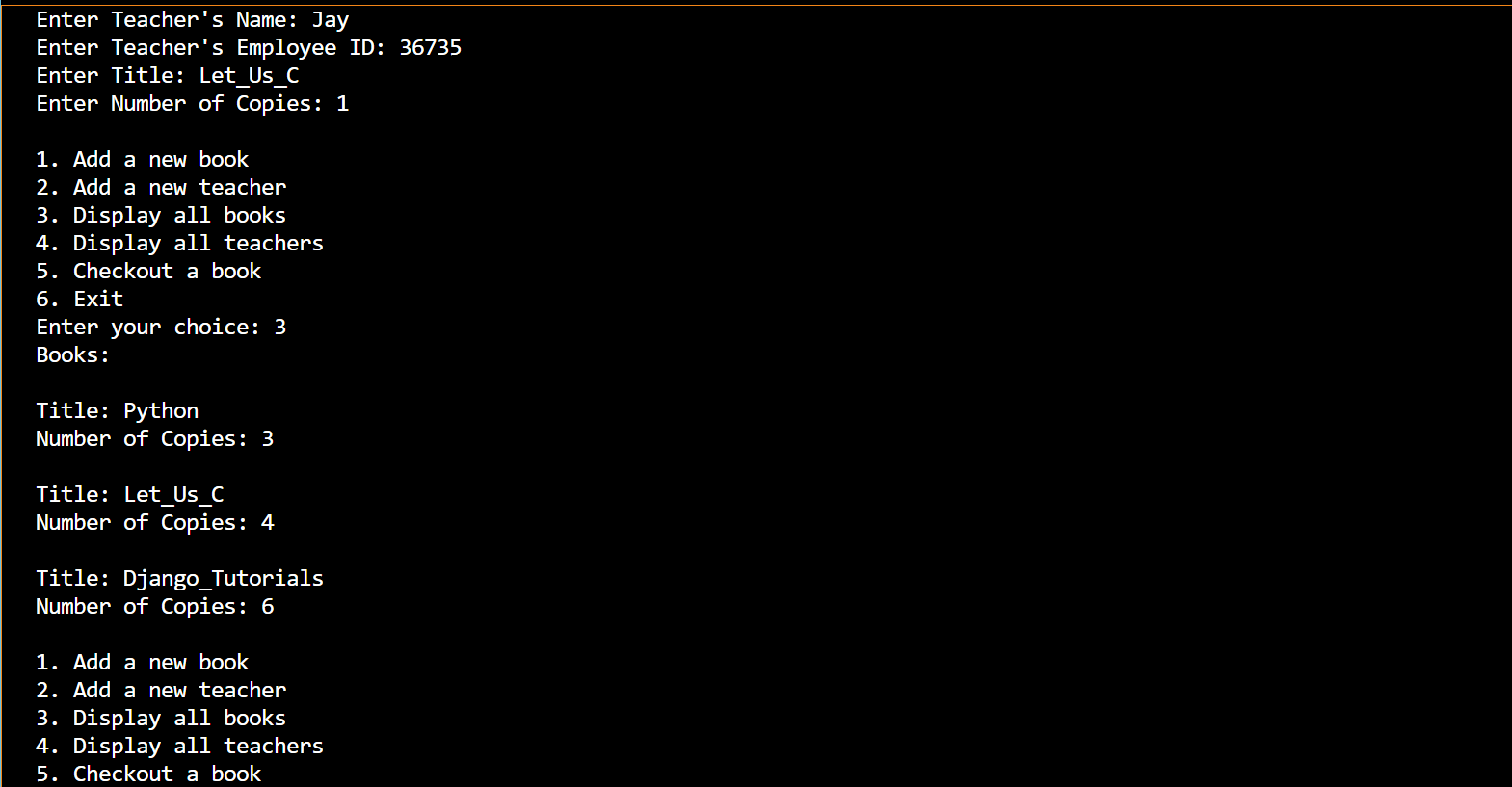
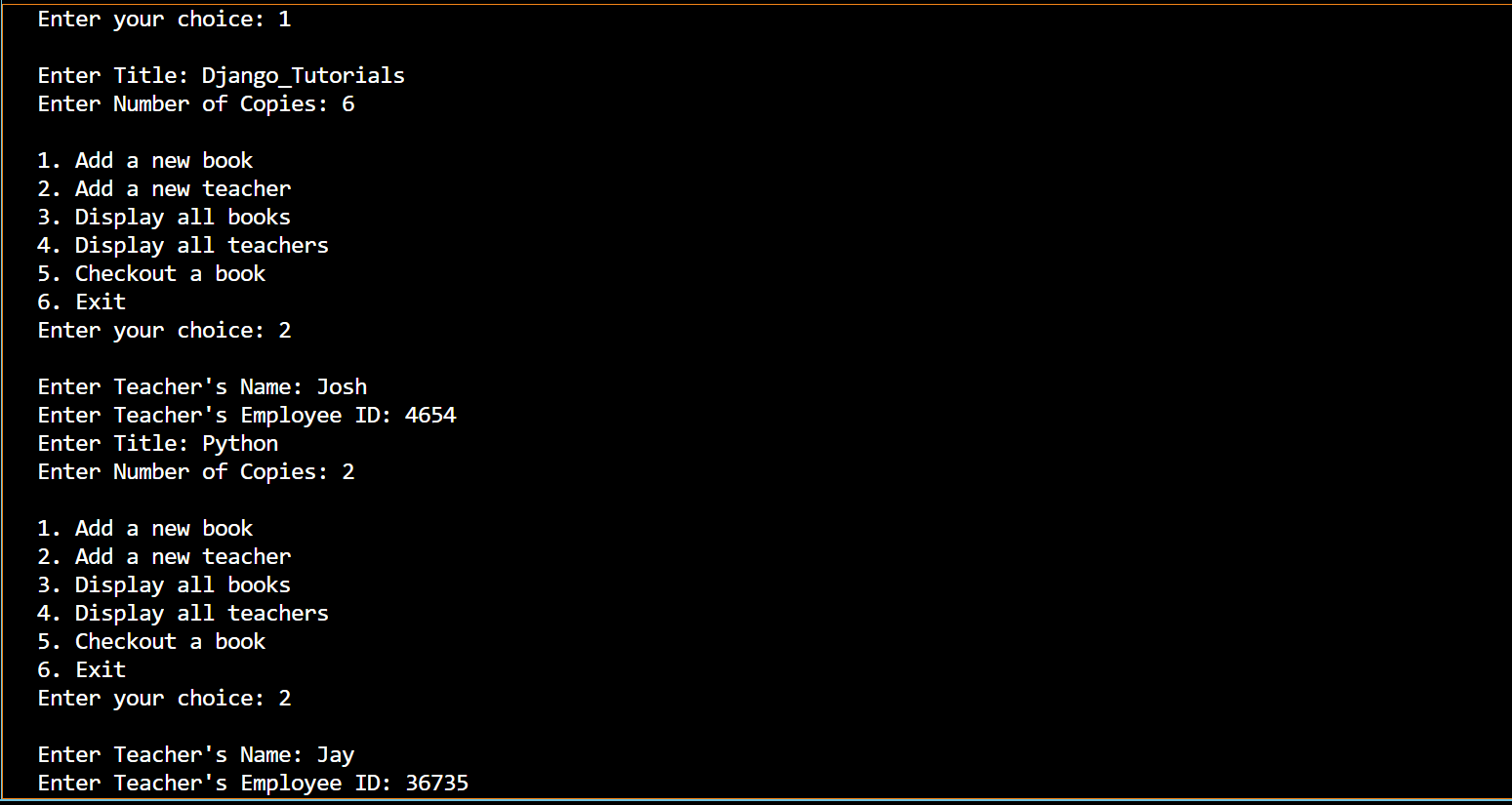
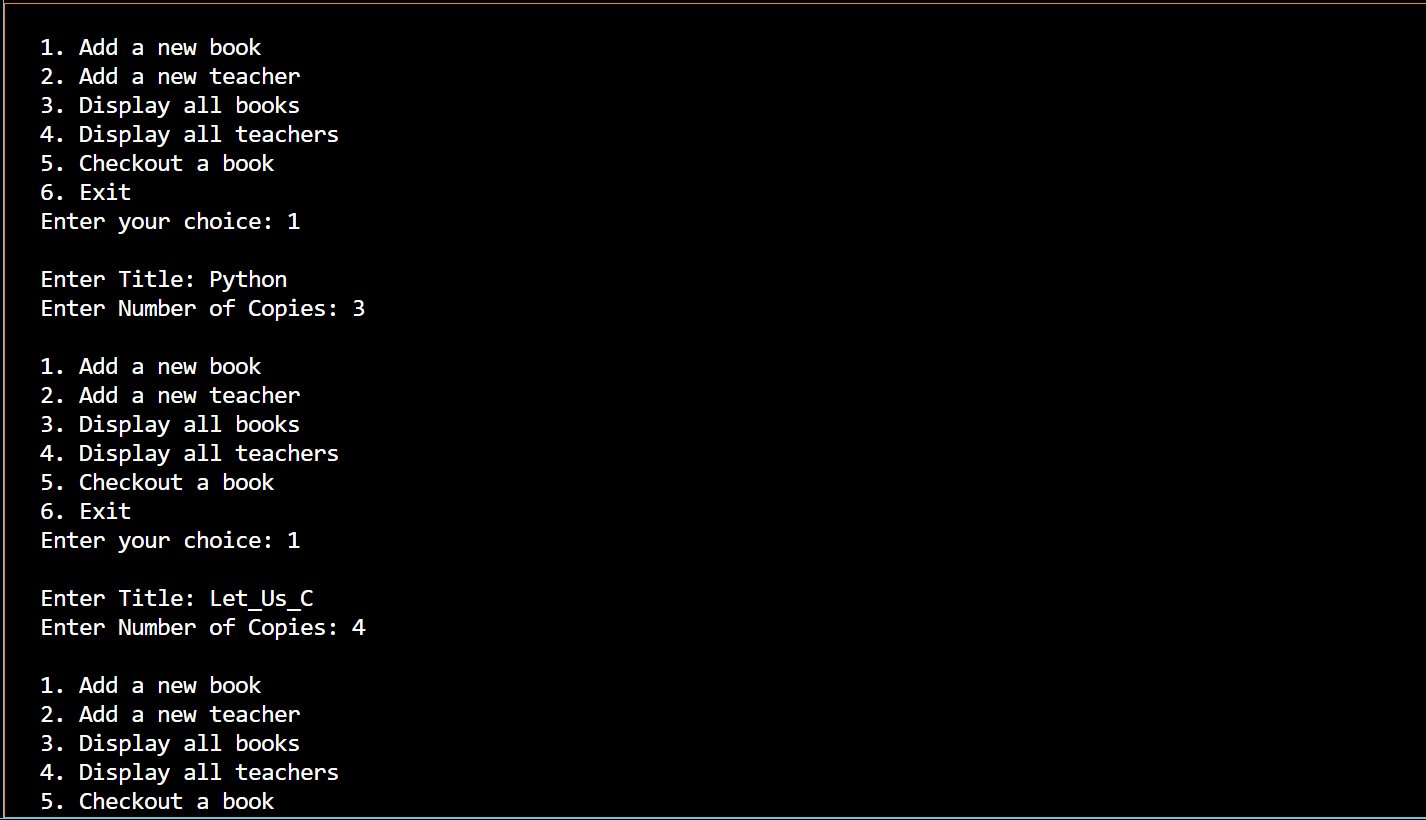
*Code snippets:*

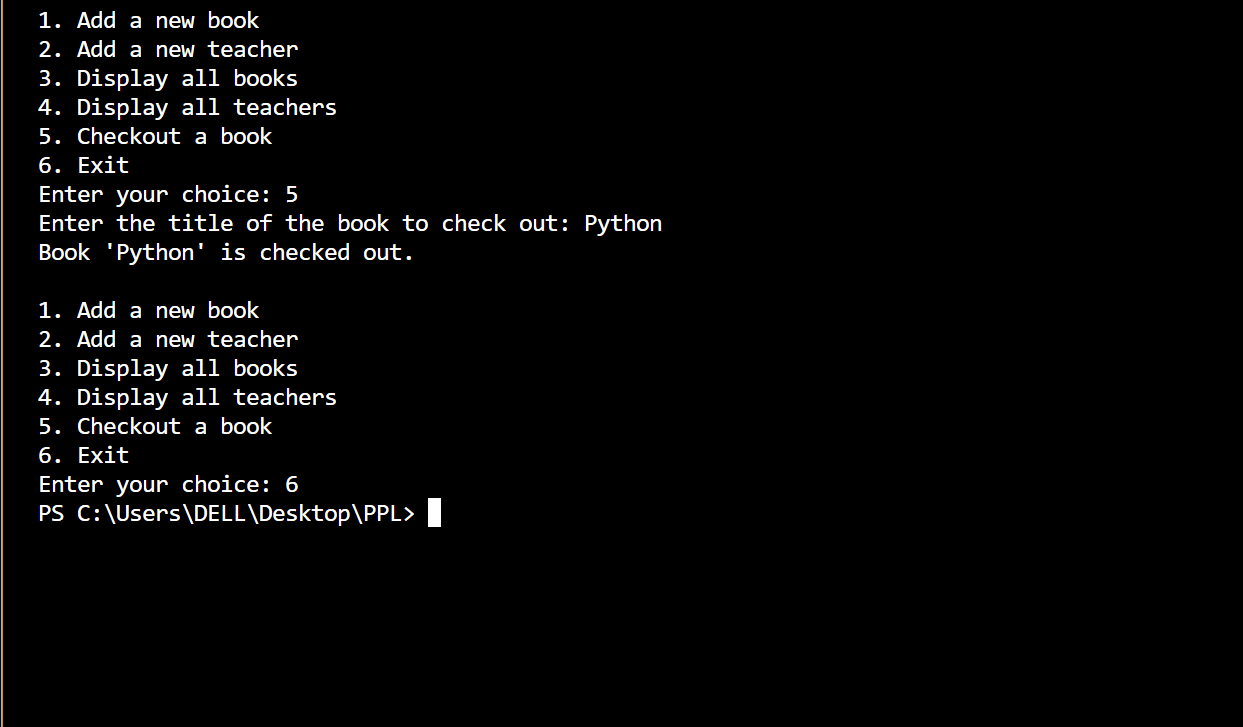




Input/Output:







**Github repository link:**

https://github.com/Janmejay-Pandya/Library-management-PP-Project.git