Web Technologies Lab

Lab 06 **Marks 100**

Instructions
Work on this lab individually.

You are **NOT** allowed to use the internet, or mobile phone.

You are **NOT** allowed to borrow anything from your peer student.

What you have to do
Program the following tasks. The name of your files will be according to the task given in this lab.

Task 1 [100]

A library wants a Java program to manage its books, storing Book ID as the key and **Book objects** (containing title, author, and availability status) as values using a **HashMap**. The system should also allow saving and loading book data using serialization.

Requirements:

1. Create a **`Book`** class that implements **`Serializable`**
   - with attributes:
     i. int bookID
     ii. String title
     iii. String author
     iv. boolean isAvailable
2. Implement a **`LibraryManager`** class with:
   - A **`HashMap<Integer, Book>`** (where `Integer` is an object type, not primitive) to store book records.
   - Methods:
     i. **`addBook(int id, String title, String author):`** Adds a new book (default: available).
     ii. **`borrowBook(int id):`** Marks a book as unavailable if it exists and is available.
     iii. **`returnBook(int id):`** Marks a book as available if it exists and is borrowed.
     iv. **`displayBooks():`** Prints all book details.
     v. **`saveToFile(String filename):`** Serializes and saves book data to a file.
     vi. **`loadFromFile`**(String filename): Deserializes and loads book data from a file.
3. In the **main method**, do the following:
   - Add multiple books using **`addBook()`**.
   - Display all books using **`displayBooks()`**.
   - Borrow a book using **`borrowBook()`**.
   - Return a book using **`returnBook()`**.
   - Save data to a file and load it back to verify serialization.

Instruction:

- Use **ObjectOutputStream** and **ObjectInputStream** for serialization.
- Handle exceptions (**IOException**, **ClassNotFoundException**) properly.
- Ensure HashMap maintains unique **Book IDs**.
- **Note:** `Integer` is an **object type**, not a primitive `int`, as `HashMap` requires objects as keys. Java automatically converts `int` to `Integer` using **autoboxing**.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺