Assignment 4 – City Library Digital Management System
Name: Azhar Ali
Roll No: 2401730134
Section: B
Faculty: Suman Vishwani

```
// ----------------------------- Book.java ------------------------------
import java.io.*;

public class Book implements Serializable {
    private Integer bookId;
    private String title;
    private String author;
    private String category;
    private boolean isIssued;

    public Book(Integer bookId, String title, String author, String category) {
        this.bookId = bookId;
        this.title = title;
        this.author = author;
        this.category = category;
        this.isIssued = false;
    }

    public void markAsIssued() { this.isIssued = true; }
    public void markAsReturned() { this.isIssued = false; }

    public void displayBookDetails() {
        System.out.println("Book ID: " + bookId);
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Category: " + category);
        System.out.println("Issued: " + (isIssued ? "Yes" : "No"));
    }

    public Integer getBookId() { return bookId; }
    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public String getCategory() { return category; }
    public boolean isIssued() { return isIssued; }
}
```

```java
// ----------------------------- Member.java ------------------------------
import java.util.*;
import java.io.*;

public class Member implements Serializable {
    private Integer memberId;
    private String name;
    private String email;
    private List<Integer> issuedBooks = new ArrayList<>();

    public Member(Integer memberId, String name, String email) {
        this.memberId = memberId;
        this.name = name;
        this.email = email;
    }

    public void addIssuedBook(int bookId) {
        issuedBooks.add(bookId);
    }

    public void returnIssuedBook(int bookId) {
        issuedBooks.remove(Integer.valueOf(bookId));
    }

    public void displayMemberDetails() {
        System.out.println("Member ID: " + memberId);
        System.out.println("Name: " + name);
        System.out.println("Email: " + email);
        System.out.println("Issued Books: " + issuedBooks);
    }

    public Integer getMemberId() { return memberId; }
    public String getName() { return name; }
    public String getEmail() { return email; }
    public List<Integer> getIssuedBooks() { return issuedBooks; }
}
```

```java
// ------------------------- LibraryManager.java (Part 1) -------------------------
import java.io.*;
import java.util.*;

public class LibraryManager {
    private Map<Integer, Book> books = new HashMap<>();
    private Map<Integer, Member> members = new HashMap<>();

    public void loadFromFile() {
        try {
            FileInputStream fis = new FileInputStream("books.dat");
            ObjectInputStream ois = new ObjectInputStream(fis);
            books = (Map<Integer, Book>) ois.readObject();
            ois.close();
        } catch (Exception e) { books = new HashMap<>(); }

        try {
            FileInputStream fis = new FileInputStream("members.dat");
            ObjectInputStream ois = new ObjectInputStream(fis);
            members = (Map<Integer, Member>) ois.readObject();
            ois.close();
        } catch (Exception e) { members = new HashMap<>(); }
    }

    public void saveToFile() {
        try {
            FileOutputStream fos = new FileOutputStream("books.dat");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(books);
            oos.close();

            fos = new FileOutputStream("members.dat");
            oos = new ObjectOutputStream(fos);
            oos.writeObject(members);
            oos.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void addBook(Book book) {
        books.put(book.getBookId(), book);
        System.out.println("Book Added Successfully!");
    }

    public void addMember(Member member) {
        members.put(member.getMemberId(), member);
        System.out.println("Member Added Successfully!");
    }
}
```

```java
// ------------------------- LibraryManager.java (Part 2) -------------------------

    public void issueBook(int memberId, int bookId) {
        if (!books.containsKey(bookId) || !members.containsKey(memberId)) {
            System.out.println("Invalid Book/Member ID!");
            return;
        }

        Book b = books.get(bookId);
        Member m = members.get(memberId);

        if (b.isIssued()) {
            System.out.println("Book already issued!");
            return;
        }

        b.markAsIssued();
        m.addIssuedBook(bookId);
        System.out.println("Book Issued Successfully!");
    }

    public void returnBook(int memberId, int bookId) {
        if (!books.containsKey(bookId) || !members.containsKey(memberId)) return;

        books.get(bookId).markAsReturned();
        members.get(memberId).returnIssuedBook(bookId);
        System.out.println("Book Returned Successfully!");
    }

    public void searchBooks(String query) {
        for (Book b : books.values()) {
            if (b.getTitle().contains(query) || b.getAuthor().contains(query) ||
                b.getCategory().contains(query))
                b.displayBookDetails();
        }
    }

    public void sortBooks() {
        List<Book> list = new ArrayList<>(books.values());
        Collections.sort(list, Comparator.comparing(Book::getTitle));

        for (Book b : list)
            b.displayBookDetails();
    }
}
```