

Case Study: Library Management System

Background

A university library wants to develop a **Library Management System** to manage book borrowing, returning, and book inventory. The system should be used by **students**, **librarians**, and **the library itself** to keep track of books.

The library follows these rules:

1. **Students** can borrow and return books.
2. **Librarians** can add and remove books from the inventory.
3. Each book has a **title, author, and ISBN number**.
4. The system should allow displaying all available books.
5. When a student borrows a book, it is removed from the library's inventory until it is returned.

Objectives

By working on this case study, you will be able to:

- ✓ **Draw class diagrams**
- ✓ **Show inheritance and relationships between classes**
- ✓ **Write a Java program that implements these classes**
- ✓ **Use variables and control structures to manage library operations**

Phase 1: Identifying Classes and Relationships

Task 1: Identify the Key Classes

- Identify at least **four** major classes required for this system.
- Define their attributes (variables) and behaviors (methods).

Questions:

1. What are the **main classes** needed for the system?
2. What are the **attributes** for each class?
3. What **methods** should each class have?

Phase 2: Designing Class Relationships and Inheritance

Task 2: Draw the Class Diagram

- Create a **UML class diagram** showing relationships between classes.
- Show **inheritance** where necessary.

Questions:

4. Which class should be the **parent class**?
5. Which classes **inherit** from the parent class?
6. What **associations** exist between classes (e.g., Library has many Books)?

Phase 3: Implementing Classes in Java

Task 3: Writing the Java Classes

- Write Java classes based on your design.
- Implement **constructors**, **getters**, and **setters** where needed.

Questions:

7. How will you define a **Book** class in Java?
8. How will you implement **inheritance** in Java?
9. What **access modifiers** should you use for class attributes?

Phase 4: Implementing Key Features

Task 4: Adding Functionality for Borrowing and Returning Books

- Implement **methods** for students to borrow and return books.
- Ensure the library updates its inventory when a book is borrowed or returned.

Questions:

10. How will the **Library class** store multiple books?
11. How will a **Student** borrow a book?
12. How will a **Student** return a book?
13. How will you check if a book is **available** before borrowing?

Phase 5: Implementing Control Structures

Task 5: Using Loops and Conditions

- Implement a **loop** to display all books in the library.
- Use an **if statement** to check if a book is available before borrowing.

Questions:

14. How will you loop through a list of books?
15. How will you check if a book exists before removing it?
16. How will you handle cases where a student tries to borrow a **non-existing book**?

Phase 6: Running the Program and Testing

Task 6: Test the Library System

- **Create a main class** to run and test the system.
- Add books, borrow books, return books, and display books.

Questions:

17. How will you **create** book objects in the main class?
18. How will you **instantiate** a student and librarian?
19. How will you test if **borrowing and returning books** work correctly?
20. How will you display books **before and after** borrowing?

Conclusion

This **case study** tests your understanding of:

- ✓ **Class diagrams and relationships**
- ✓ **Inheritance in Java**
- ✓ **Using variables and methods**
- ✓ **Control structures (loops, conditions)**
- ✓ **Object-oriented programming best practices**