Java Collection Assignments

Assignment 1: Employee Management System (CRUD + Searching by ID)

Scenario:
You are building a small Employee Management System for an IT company. Employees are stored in an ArrayList<Employee>.

Requirements:

1. Create an Employee class with the following fields:
    - int id
    - String name
    - String department
    - double salary
2. Implement equals() and hashCode() in such a way that **two employees are considered equal if their id is the same**.
3. Create a main class to perform the following operations using ArrayList:
    - **Add Employee** (CRUD - Create)
    - **Update Employee salary or department** based on id (CRUD - Update)
    - **Delete Employee** based on id (CRUD - Delete)
    - **Search Employee** by id using contains() and indexOf() methods (uses overridden equals)
    - **Display all employees** (CRUD - Read)

👉 *Hint:* Use a loop for update/delete operations.

Assignment 2: Library Book Management (Sorting + Comparator)

Scenario:
A library stores all its books in an ArrayList<Book>. The librarian wants to sort books in different ways depending on the requirement.

Requirements:

1. Create a Book class with:
    - int bookId
    - String title
    - String author
    - double price

2. Implement **Comparable** to sort books **by title (alphabetical order)**.
3. Create multiple **Comparator** classes for sorting:
    - By price (ascending order)
    - By author (alphabetical order)
4. In main, demonstrate:
    - Adding at least 5 books into an ArrayList
    - Sorting by title (Comparable)
    - Sorting by price and author (Comparator)
    - Printing the results after each sorting

Assignment 3: Online Shopping Cart (CRUD + Searching)

Scenario:
An online shopping cart system keeps track of items selected by a user. Items are stored in an ArrayList<CartItem>.

Requirements:

1. Create a CartItem class with:
    - int itemId
    - String itemName
    - int quantity
    - double pricePerUnit
2. Implement equals() and hashCode() so that two CartItem objects are equal if their itemId is the same.
3. In the ShoppingCart class, implement methods:
    - addItem(CartItem item) – Add item (if already present, just increase quantity instead of adding duplicate).
    - removeItem(int itemId) – Remove an item.
    - updateQuantity(int itemId, int newQuantity) – Update item quantity.
    - searchItem(int itemId) – Search by itemId.
    - getTotalBill() – Return the total cost of items in the cart.

👉 *Hint:* Use equals() in contains() to check if an item already exists before adding.

Assignment 4: Student Result Management (Sorting + Searching)

Scenario:
A university maintains a list of students and their scores in an ArrayList<Student>.

Requirements:

1. Create a Student class with:
    - int rollNo
    - String name
    - double marks
2. Implement Comparable to sort students by **marks (descending order)**.
3. Implement at least two Comparator classes:
    - Sort by name (alphabetical order)
    - Sort by rollNo (ascending order)
4. Implement CRUD operations in main:
    - Add new students
    - Delete a student by roll number
    - Update marks by roll number
    - Search for a student by roll number (using equals)
5. Show sorting results by different criteria.

✅ These 4 assignments cover:

- **CRUD Operations with List/ArrayList**
- **Searching with equals() and hashCode()**
- **Sorting using Comparable and Comparator**
- **Real-world relatable industry scenarios**