**Exercise 1) - Easy**

You are tasked with creating a simple system to manage and organize a list of books in a small library. Each book has the following attributes: title (String), author (String), and yearPublished (int). Your task is to:

1. Create a Book class with the above attributes.
2. Override the equals method to compare books based on their title and author. Two books are considered the same if they have the same title and author, even if their publication years differ.
3. Implement the Comparable interface to allow sorting the books by title in alphabetical order.
4. Override the toString method to provide a readable string representation of a book.
5. In the main method, create an ArrayList of Book objects. Add at least five books to the list.
6. Sort the list of books using Collections.sort() and print the sorted list.
7. Test the equals method by checking if two books with the same title and author but different years are considered equal.

**Solution**:

import java.util.ArrayList;

import java.util.Collections;

class Book implements Comparable<Book> {

    String title;

    String author;

    int yearPublished;

    Book(String title, String author, int yearPublished) {

        this.title = title;

        this.author = author;

        this.yearPublished = yearPublished;

    }

    // Overriding equals to compare title and author

    @Override

    public boolean equals(Object o) {

        if (this == o) return true;

        if (o == null || getClass() != o.getClass()) return false;

        Book book = (Book) o;

        return title.equals(book.title) && author.equals(book.author);

    }

    // Implementing the Comparable interface to sort by title

    @Override

    public int compareTo(Book other) {

        return this.title.compareTo(other.title);

    }

    // Overriding toString to give a readable representation of the book

    @Override

    public String toString() {

        return "Book{" +

                "title='" + title + '\'' +

                ", author='" + author + '\'' +

                ", yearPublished=" + yearPublished +

                '}';

    }

}

public class Main {

    public static void main(String[] args) {

        ArrayList<Book> books = new ArrayList<>();

        // Adding books to the ArrayList

        books.add(new Book("The Great Gatsby", "F. Scott Fitzgerald", 1925));

        books.add(new Book("Moby Dick", "Herman Melville", 1851));

        books.add(new Book("1984", "George Orwell", 1949));

        books.add(new Book("To Kill a Mockingbird", "Harper Lee", 1960));

        books.add(new Book("The Great Gatsby", "F. Scott Fitzgerald", 1992)); // Same title & author, different year

        // Sorting books by title

        Collections.sort(books);

        // Printing sorted list

        System.out.println("Books sorted by title:");

        for (Book book : books) {

            System.out.println(book);

        }

        // Checking equality of two books with the same title and author but different years

        Book book1 = new Book("The Great Gatsby", "F. Scott Fitzgerald", 1925);

        Book book2 = new Book("The Great Gatsby", "F. Scott Fitzgerald", 1992);

        System.out.println("\nAre book1 and book2 equal? " + book1.equals(book2));

    }

}

**Exercise 2) – Easy**

In this exercise, you will expand upon the previous book management system by adding functionality to sort the books by the year they were published, in addition to sorting by title. You will:

1. Modify the Book class by creating a static method sortByYear that uses a custom Comparator to sort the list of books by their publication year in ascending order.
2. In the main method, after sorting the books by title, use this new method to sort the books by their publication year.
3. Print the list of books sorted by both title and year.

**Solution:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

class Book implements Comparable<Book> {

    String title;

    String author;

    int yearPublished;

    Book(String title, String author, int yearPublished) {

        this.title = title;

        this.author = author;

        this.yearPublished = yearPublished;

    }

    @Override

    public boolean equals(Object o) {

        if (this == o) return true;

        if (o == null || getClass() != o.getClass()) return false;

        Book book = (Book) o;

        return title.equals(book.title) && author.equals(book.author);

    }

    @Override

    public int compareTo(Book other) {

        return this.title.compareTo(other.title);

    }

    @Override

    public String toString() {

        return "Book{" +

                "title='" + title + '\'' +

                ", author='" + author + '\'' +

                ", yearPublished=" + yearPublished +

                '}';

    }

    // Static method to sort books by year using a custom Comparator

    public static void sortByYear(ArrayList<Book> books) {

        Collections.sort(books, new Comparator<Book>() {

            @Override

            public int compare(Book b1, Book b2) {

                return Integer.compare(b1.yearPublished, b2.yearPublished);

            }

        });

    }

}

public class Main {

    public static void main(String[] args) {

        ArrayList<Book> books = new ArrayList<>();

        books.add(new Book("The Great Gatsby", "F. Scott Fitzgerald", 1925));

        books.add(new Book("Moby Dick", "Herman Melville", 1851));

        books.add(new Book("1984", "George Orwell", 1949));

        books.add(new Book("To Kill a Mockingbird", "Harper Lee", 1960));

        books.add(new Book("The Great Gatsby", "F. Scott Fitzgerald", 1992));

        // Sort books by title

        Collections.sort(books);

        System.out.println("Books sorted by title:");

        for (Book book : books) {

            System.out.println(book);

        }

        // Sort books by year using the static method

        Book.sortByYear(books);

        System.out.println("\nBooks sorted by year:");

        for (Book book : books) {

            System.out.println(book);

        }

    }

}

**Exercise 3) – Medium**

In this exercise, you will extend the functionality of the book management system by adding the ability to sort books based on multiple criteria and allow for dynamic sorting. Specifically, you will:

1. Modify the Book class to include a new genre attribute (String).
2. Implement the ability to dynamically sort books based on any combination of attributes: title, yearPublished, or genre.
3. Create a method sortBooks that takes an ArrayList<Book> and two boolean flags: sortByTitleFirst and sortByGenreFirst.
   * If sortByTitleFirst is true, books should be sorted first by title, then by year.
   * If sortByGenreFirst is true, books should be sorted first by genre, then by year.
   * If both flags are false, sort only by year.
4. In the main method, prompt the user (via console input) to choose the sorting method and demonstrate the sorting based on the user’s selection.

**Solution:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.Scanner;

class Book implements Comparable<Book> {

    String title;

    String author;

    int yearPublished;

    String genre;

    Book(String title, String author, int yearPublished, String genre) {

        this.title = title;

        this.author = author;

        this.yearPublished = yearPublished;

        this.genre = genre;

    }

    @Override

    public boolean equals(Object o) {

        if (this == o) return true;

        if (o == null || getClass() != o.getClass()) return false;

        Book book = (Book) o;

        return title.equals(book.title) && author.equals(book.author);

    }

    @Override

    public int compareTo(Book other) {

        return this.title.compareTo(other.title);

    }

    @Override

    public String toString() {

        return "Book{" +

                "title='" + title + '\'' +

                ", author='" + author + '\'' +

                ", yearPublished=" + yearPublished +

                ", genre='" + genre + '\'' +

                '}';

    }

    // Static method to dynamically sort books based on flags

    public static void sortBooks(ArrayList<Book> books, boolean sortByTitleFirst, boolean sortByGenreFirst) {

        Collections.sort(books, new Comparator<Book>() {

            @Override

            public int compare(Book b1, Book b2) {

                if (sortByTitleFirst) {

                    int titleComparison = b1.title.compareTo(b2.title);

                    if (titleComparison != 0) return titleComparison;

                    return Integer.compare(b1.yearPublished, b2.yearPublished);

                } else if (sortByGenreFirst) {

                    int genreComparison = b1.genre.compareTo(b2.genre);

                    if (genreComparison != 0) return genreComparison;

                    return Integer.compare(b1.yearPublished, b2.yearPublished);

                } else {

                    return Integer.compare(b1.yearPublished, b2.yearPublished);

                }

            }

        });

    }

}

public class Main {

    public static void main(String[] args) {

        ArrayList<Book> books = new ArrayList<>();

        books.add(new Book("The Great Gatsby", "F. Scott Fitzgerald", 1925, "Fiction"));

        books.add(new Book("Moby Dick", "Herman Melville", 1851, "Adventure"));

        books.add(new Book("1984", "George Orwell", 1949, "Dystopian"));

        books.add(new Book("To Kill a Mockingbird", "Harper Lee", 1960, "Fiction"));

        books.add(new Book("Brave New World", "Aldous Huxley", 1932, "Dystopian"));

        // Get user input for sorting preference

        Scanner scanner = new Scanner(System.in);

        System.out.println("Choose sorting method:");

        System.out.println("1. Sort by Title first, then by Year");

        System.out.println("2. Sort by Genre first, then by Year");

        System.out.println("3. Sort by Year only");

        int choice = scanner.nextInt();

        // Determine sorting method based on user input

        boolean sortByTitleFirst = (choice == 1);

        boolean sortByGenreFirst = (choice == 2);

        // Sort the books based on user choice

        Book.sortBooks(books, sortByTitleFirst, sortByGenreFirst);

        // Print sorted books

        System.out.println("\nSorted books:");

        for (Book book : books) {

            System.out.println(book);

        }

    }

}