

Exercise#01: Comparable Persons

Consider the following **Person** class that implements the **Comparable** interface:

```
public class Person implements Comparable<Person> {  
    private String name;  
    private int age;  
  
    // Constructor and getters/setters needed  
  
    public int compareTo(Person other) {  
        // Implement the compareTo method to compare persons based on age  
        // Return a negative value if 'this' is less than 'other', 0 if equal, and a positive value if 'this' is greater than  
        'other'  
    }  
  
    public static void main(String[] args) {  
        // Create an ArrayList of Person objects and demonstrate sorting using Collections.sort  
    }  
}
```

Instructions:

1. Complete the *compareTo* Method:

- Inside the **compareTo** method, compare two **Person** objects based on their ages.
- Return a negative value if the age of the current person (**this**) is less than the age of the other person (**other**).
- Return 0 if the ages are equal.
- Return a positive value if the age of the current person is greater than the age of the other person.

2. Demonstrate Sorting:

- In the **main** method, create an **ArrayList** of **Person** objects with various names and ages.
- Use **Collections.sort** to sort the list of persons based on their ages.
- Print the sorted list to demonstrate that the persons are now arranged in ascending order of age.

Note:

- Ensure that your **compareTo** method provides a consistent and meaningful way to compare **Person** objects based on their ages.
- Test your implementation by creating a variety of **Person** objects and sorting them using the provided **main** method.

Exercise#02: Comparator for Books

Consider the following **Book** class that implements the **Comparable** interface and includes two **Comparator** instances:

```
public class Book implements Comparable<Book> {
    private String title;
    private String author;
    private int year;

    // Constructor and getters/setters

    public static Comparator<Book> yearComparator = new Comparator<Book>() {
        // Implement the compare method to compare books based on the publication year
    };

    public static Comparator<Book> authorComparator = new Comparator<Book>() {
        // Implement the compare method to compare books based on the author's name
    };

    public static void main(String[] args) {
        // Create an array of Book objects and demonstrate sorting using Arrays.sort with different
        // comparators
    }
}
```

Instructions:

1. Complete yearComparator:

- Inside the **yearComparator**, implement the **compare** method to compare two **Book** objects based on their publication years.
- Return a negative value if the publication year of the first book (**this**) is earlier than the second book (**other**).
- Return 0 if the publication years are equal.
- Return a positive value if the publication year of the first book is later than the second book.

2. Complete authorComparator:

- Inside the **authorComparator**, implement the **compare** method to compare two **Book** objects based on their author names.

- Use the **String** class's **compareTo** method to perform the comparison.
- Return a negative value if the author's name of the first book is lexicographically less than the second book.
- Return 0 if the author names are lexicographically equal.
- Return a positive value if the author's name of the first book is lexicographically greater than the second book.

3. **Demonstrate Sorting:**

- In the **main** method, create an array of **Book** objects with various titles, authors, and publication years.
- Use **Arrays.sort** to sort the array of books using both the **yearComparator** and **authorComparator**.
- Print the sorted arrays to demonstrate that the books are now arranged based on either publication year or author's name.

Note:

- Ensure that your comparators provide consistent and meaningful ways to compare **Book** objects based on publication year and author's name.
- Test your implementation by creating a variety of **Book** objects and sorting them using **Arrays.sort** with both comparators.