Assignment Questions on Custom Sorting with Comparable

1. **Sorting Students by Grade**:

   Write a Java program to create a Student class with properties id, name, and grade. Implement the Comparable interface to sort students by their grades in ascending order. Create an ArrayList of Student objects, add five students to the list, and display the sorted list.

2. **Sorting Employees by Salary**:

   Write a Java program to create an Employee class with properties id, name, and salary. Implement the Comparable interface to sort employees by their salaries in descending order. Create a LinkedList of Employee objects, add five employees to the list, and display the sorted list.

3. **Sorting Books by Title**:

   Write a Java program to create a Book class with properties isbn, title, and author. Implement the Comparable interface to sort books by their titles in alphabetical order. Create a Vector of Book objects, add five books to the vector, and display the sorted list.

4. **Sorting Movies by Release Year**:

   Write a Java program to create a Movie class with properties title, genre, and year. Implement the Comparable interface to sort movies by their release years in ascending order. Create a Stack of Movie objects, add five movies to the stack, and display the sorted list.

5. **Sorting Products by Price**:

   Write a Java program to create a Product class with properties productId, name, and price. Implement the Comparable interface to sort products by their prices in ascending order. Create an ArrayList of Product objects, add five products to the list, and display the sorted list.

6. **Sorting Customers by Age**:

   Write a Java program to create a Customer class with properties customerId, name, and age. Implement the Comparable interface to sort customers by their ages in ascending order. Create a LinkedList of Customer objects, add five customers to the list, and display the sorted list.

7. **Sorting Courses by Duration**:

Write a Java program to create a Course class with properties courseId, courseName, and duration (in hours). Implement the Comparable interface to sort courses by their duration in ascending order. Create a Vector of Course objects, add five courses to the vector, and display the sorted list.

8. **Sorting Orders by Amount**:

   Write a Java program to create an Order class with properties orderId, product, and amount. Implement the Comparable interface to sort orders by their amounts in ascending order. Create an ArrayList of Order objects, add five orders to the list, and display the sorted list.

9. **Sorting Events by Date**:

   Write a Java program to create an Event class with properties eventId, eventName, and eventDate (use LocalDate). Implement the Comparable interface to sort events by their dates in ascending order. Create a LinkedList of Event objects, add five events to the list, and display the sorted list.

10. **Sorting Projects by Deadline**:

    Write a Java program to create a Project class with properties projectId, projectName, and deadline (use LocalDate). Implement the Comparable interface to sort projects by their deadlines in ascending order. Create an ArrayList of Project objects, add five projects to the list, and display the sorted list.

11. **Sorting Patients by Admission Date**:

    Write a Java program to create a Patient class with properties patientId, name, and admissionDate (use LocalDate). Implement the Comparable interface to sort patients by their admission dates in ascending order. Create a Vector of Patient objects, add five patients to the vector, and display the sorted list.

12. **Sorting Flights by Departure Time**:

    Write a Java program to create a Flight class with properties flightId, destination, and departureTime (use LocalTime). Implement the Comparable interface to sort flights by their departure times in ascending order. Create a Stack of Flight objects, add five flights to the stack, and display the sorted list.

13. **Sorting Tasks by Priority**:

    Write a Java program to create a Task class with properties taskId, description, and priority (use an enum with values LOW, MEDIUM, HIGH). Implement the Comparable interface to sort tasks by their priority in ascending order (LOW < MEDIUM <

HIGH). Create an ArrayList of Task objects, add five tasks to the list, and display the sorted list.

14. **Sorting Books by Publication Year**:

    Write a Java program to create a Book class with properties isbn, title, and publicationYear. Implement the Comparable interface to sort books by their publication years in ascending order. Create a LinkedList of Book objects, add five books to the list, and display the sorted list.

15. **Sorting Meetings by Start Time**:

    Write a Java program to create a Meeting class with properties meetingId, topic, and startTime (use LocalTime). Implement the Comparable interface to sort meetings by their start times in ascending order. Create an ArrayList of Meeting objects, add five meetings to the list, and display the sorted list.