**Exercise 3: Sorting Customer Orders**

**Main.java**

package sortingcustomerorder;

public class Main {

    public static void main(String[] args) {

        Order[] orders = {

            new Order("O1", "Bharath", 2500),

            new Order("O2", "Geethu", 1500),

            new Order("O3", "Ganesh", 3500),

            new Order("O4", "Ramesh", 2000),

            new Order("O5", "Suresh", 3000),

            new Order("O6", "Anil", 1800),

            new Order("O7", "Kiran", 4000),

            new Order("O8", "Priya", 2200)

        };

        System.out.println("Original Orders:");

        for (Order o : orders) System.out.println(o);

        // Bubble Sort

        OrderSorter.bubbleSort(orders);

        System.out.println("\nOrders after Bubble Sort:");

        for (Order o : orders) System.out.println(o);

        // Shuffle for demonstration (or recreate array)

        orders = new Order[]{

            new Order("O1", "Alice", 2500),

            new Order("O2", "Bob", 1500),

            new Order("O3", "Charlie", 3500),

            new Order("O4", "Diana", 2000)

        };

        // Quick Sort

        OrderSorter.quickSort(orders, 0, orders.length - 1);

        System.out.println("\nOrders after Quick Sort:");

        for (Order o : orders) System.out.println(o);

    }

}

**Order.java**

package sortingcustomerorder;

public class Order {

    private String orderId;

    private String customerName;

    private double totalPrice;

    public Order(String orderId, String customerName, double totalPrice) {

        this.orderId = orderId;

        this.customerName = customerName;

        this.totalPrice = totalPrice;

    }

    public String getOrderId() { return orderId; }

    public String getCustomerName() { return customerName; }

    public double getTotalPrice() { return totalPrice; }

    @Override

    public String toString() {

        return "OrderID: " + orderId + ", Customer: " + customerName + ", Total: " + totalPrice;

    }

}

**OrderSorter.java**

package sortingcustomerorder;

public class OrderSorter {

    // Bubble Sort

    public static void bubbleSort(Order[] orders) {

        int n = orders.length;

        for (int i = 0; i < n-1; i++) {

            for (int j = 0; j < n-i-1; j++) {

                if (orders[j].getTotalPrice() > orders[j+1].getTotalPrice()) {

                    Order temp = orders[j];

                    orders[j] = orders[j+1];

                    orders[j+1] = temp;

                }

            }

        }

    }

    // Quick Sort

    public static void quickSort(Order[] orders, int low, int high) {

        if (low < high) {

            int pi = partition(orders, low, high);

            quickSort(orders, low, pi - 1);

            quickSort(orders, pi + 1, high);

        }

    }

    private static int partition(Order[] orders, int low, int high) {

        double pivot = orders[high].getTotalPrice();

        int i = (low - 1);

        for (int j = low; j < high; j++) {

            if (orders[j].getTotalPrice() < pivot) {

                i++;

                Order temp = orders[i];

                orders[i] = orders[j];

                orders[j] = temp;

            }

        }

        Order temp = orders[i + 1];

        orders[i + 1] = orders[high];

        orders[high] = temp;

        return i + 1;

    }

}

