**Exercise 3: Sorting Customers Orders**

**Order.java**

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 double getTotalPrice() {

return totalPrice;

}

public String toString() {

return "[" + orderId + "] " + customerName + " - ₹" + totalPrice;

}

}

**BubbleSort.java**

public class BubbleSort {

public static void sort(Order[] orders) {

int n = orders.length;

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

boolean swapped = false;

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

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

Order temp = orders[j];

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

orders[j + 1] = temp;

swapped = true;

}

}

if (!swapped) break;

}

}

}

QuickSort.java

public class QuickSort {

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

if (low < high) {

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

*sort*(orders, low, pi - 1);

*sort*(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;

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

Order[] orders = {

new Order("O101", "Michael", 4500),

new Order("O102", "Babu", 1800),

new Order("O103", "CharuLatha", 4000),

new Order("O104", "Dhinesh", 1700)

};

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

*printOrders*(orders);

// Bubble Sort

System.***out***.println("\n Sorted by Bubble Sort (Ascending):");

BubbleSort.*sort*(orders);

*printOrders*(orders);

// Reset data

orders = new Order[] {

new Order("O101", "Michael", 4500),

new Order("O102", "Babu", 1800),

new Order("O103", "CharuLatha", 4000),

new Order("O104", "Dhinesh", 1700)

};

// Quick Sort

System.***out***.println("\n⚡ Sorted by Quick Sort (Ascending):");

QuickSort.*sort*(orders, 0, orders.length - 1);

*printOrders*(orders);

}

public static void printOrders(Order[] orders) {

for (Order order : orders) {

System.***out***.println(order);

}

}

}

**Output:**

