***Java Programming***

***section-3.2 practice***

**Java Program for Sorting and Searching:**

public class SortAndSearch {

public static void main(String[] args) {

// b. Create an integer array named numbers that will hold 50 values.

int[] numbers = new int[50];

// c. Fill the array with random integer values between 0 and 100.

fillArray(numbers);

// d. Display the contents of the array under the heading "Unordered list".

System.out.println("Unordered list:");

displayArray(numbers);

// e. Get the number to be searched for from the user.

java.util.Scanner scanner = new java.util.Scanner(System.in);

System.out.print("Enter a number to search for: ");

int searchNumber = scanner.nextInt();

// f. Use a sequential/linear search to identify if the value is in the array.

int index = linearSearch(numbers, searchNumber);

if (index != -1) {

System.out.println("Number found at position: " + index);

} else {

System.out.println("Number not found.");

}

// h. Sort the array using a bubble sort.

bubbleSort(numbers);

// i. Display the contents of the array under the heading "Ordered list".

System.out.println("Ordered list:");

displayArray(numbers);

// j. Use a sequential/linear search again to find the number.

index = linearSearch(numbers, searchNumber);

if (index != -1) {

System.out.println("Number found at position: " + index);

} else {

System.out.println("Number not found.");

}

}

// Method to fill the array with random values between 0 and 100

private static void fillArray(int[] array) {

java.util.Random rand = new java.util.Random();

for (int i = 0; i < array.length; i++) {

array[i] = rand.nextInt(101); // random number between 0 and 100

}

}

// Method to display the contents of the array

private static void displayArray(int[] array) {

for (int num : array) {

System.out.print(num + " ");

}

System.out.println();

}

// Method for linear search

private static int linearSearch(int[] array, int key) {

for (int i = 0; i < array.length; i++) {

if (array[i] == key) {

return i; // found

}

}

return -1; // not found

}

// Method for bubble sort

private static void bubbleSort(int[] array) {

int n = array.length;

boolean swapped;

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

swapped = false;

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

if (array[j] > array[j + 1]) {

// Swap array[j] and array[j + 1]

int temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

swapped = true;

}

}

// If no elements were swapped, break

if (!swapped) break;

}

}

}

