Laboratory Assignment File

for

Data Structure and Algorithms

Master of Technology

in

Computer Science & Engineering

Submitted by

Yugal

(Roll no. 25903053)

Department of Computer Science & Engineering
Dr. B R Ambedkar National Institute of Technology Jalandhar
Punjab, India-144008
May, 2026

Contents

Assignment 1: Basic Programs 28-01-2026 1

1.1 Hello World 1

1.2 Write a program to show Array 1

1.3 Write a program to show sum and average 1

Assignment 2: Sorting Algorithms 28-01-2026 3

2.1 Write a program to show Bubble Sort 3

2.2 Write a program to show Insertion Sort 3

2.3 Write a program to show Selection Sort 4

2.4 Write a program to show Merge Sort 5

2.5 Write a program to show Quick Sort 6

List of Figures

1 Program to show Array 1
2 Program to show sum and average 2
3 Program to show Bubble Sort 3
4 Program to show Insertion Sort 4
5 Program to show Selection Sort 5
6 Program to show Merge Sort 6
7 Program to show Quick Sort 8

Assignment 1: Basic Programs

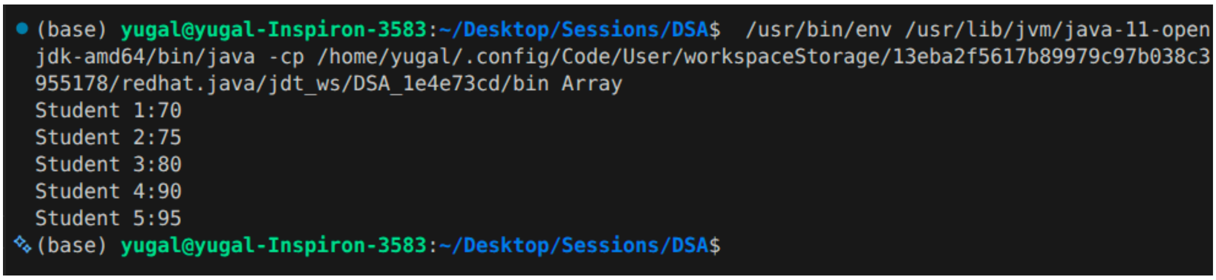
1.1 Hello World

```java
public class Hello{
    public static void main(String arg[]){
    System.out.println("Hello");
    }
}
```

1.2 Write a program to show Array

```java
// Store and display the marks
public class Array{
    public static void main(String... args){
        int[] marks = {70,75,80,90,95};
        for(int i=0;i<marks.length;i++){
            // System.out.println("Student Marks: "+marks[i]);
            System.out.println("Student " + (i+1) +":"+marks[i]);
        }
    }

}
```

Output



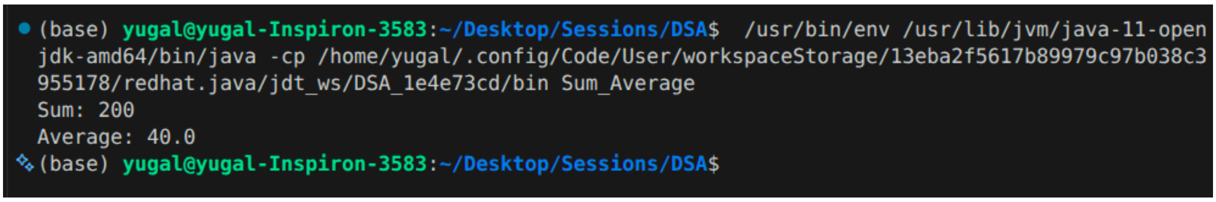
```
● (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
  jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
  955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin Array
  Student 1:70
  Student 2:75
  Student 3:80
  Student 4:90
  Student 5:95
  (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 1: Program to show Array

1.3 Write a program to show sum and average

```java
// Sum and average of array elements
class Sum_Average {

    public static void main(String[] args){
        int[] a = {20,30,40,50,60};
        int sum = 0;

        for(int x:a){
            sum += x;
        }
        double average = (double) sum/a.length;

        System.out.println("Sum: "+ sum);
        System.out.println("Average: "+ average);
    }
}
```

Output



```
● (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
  jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
  955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin Sum_Average
  Sum: 200
  Average: 40.0
 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

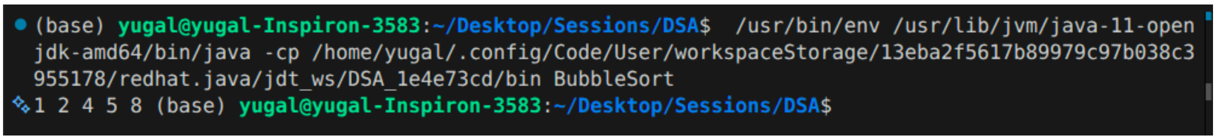
Figure 2: Program to show sum and average

Assignment 2: Sorting Algorithms

2.1 Write a program to show Bubble Sort

```java
// Sorting of array in Bubble Sort
public class BubbleSort{
    public static void main(String[] args){
        int arr[] = {5,1,4,2,8};
        for(int i=0; i<arr.length-1; i++){
            for(int j=0; j<arr.length-i-1; j++){
                if(arr[j]>arr[j+1]){
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }
        for(int num : arr){
            System.out.print(num + " ");
        }
    }
}
```

Output



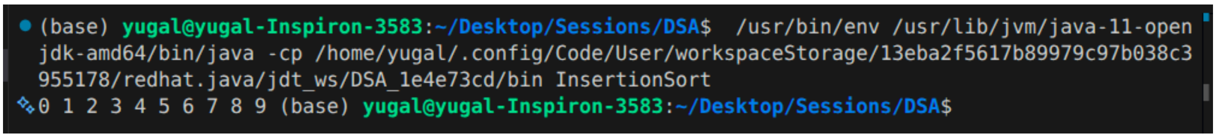
```
● (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
  jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
  955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin BubbleSort
1 2 4 5 8 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 3: Program to show Bubble Sort

2.2 Write a program to show Insertion Sort

```java
// Insertion Sort
public class InsertionSort {
    public static void main(String[] args) {
        int arr[] = {1,4,5,2,3,7,9,8,0,6};
        for(int i = 1;i<arr.length; i++){
            int key = arr[i];
            int j = i-1;
            while (j>=0 && arr[j]>key) {
                arr[j+1] = arr[j];
                j--;
            }
            arr[j+1] = key;
        }
        for(int num : arr){
            System.out.print(num + " ");
        }
    }
}
```

Output



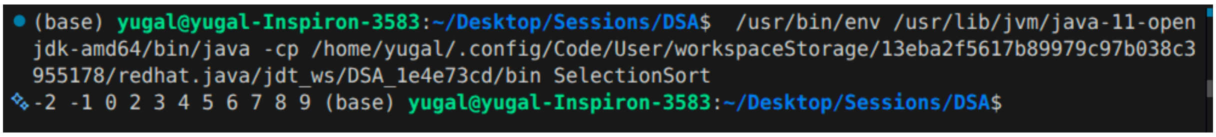
```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin InsertionSort
0 1 2 3 4 5 6 7 8 9 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 4: Program to show Insertion Sort

2.3 Write a program to show Selection Sort

```
    //Selection Sort
public class SelectionSort {
    public static void main(String[] args) {
        int arr[] = {3,5,2,6,-1,0,4,7,9,8,-2};

        for(int i=0;i<arr.length-1;i++){
            int minIndex = i;
            for(int j = i+1; j<arr.length;j++){
                if(arr[j]<arr[minIndex]){
                    minIndex = j;

                }
            }
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }
        for(int num : arr){
            System.out.print(num + " ");

        }
    }

}
```

Output



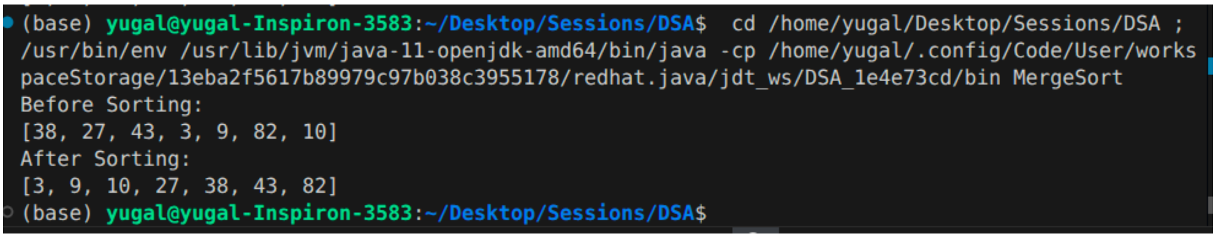
```
● (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  /usr/bin/env /usr/lib/jvm/java-11-open
  jdk-amd64/bin/java -cp /home/yugal/.config/Code/User/workspaceStorage/13eba2f5617b89979c97b038c3
  955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin SelectionSort
-2 -1 0 2 3 4 5 6 7 8 9 (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 5: Program to show Selection Sort

2.4 Write a program to show Merge Sort

```java
    //Merge Sort : Divide and Conquer
import java.util.Arrays;

public class MergeSort {

    // Main merge sort function
    public static void mergeSort(int[] arr, int left, int right){
        if(left < right){
            int mid =  left + (right-left)/2;
            // Sort Left Half
            mergeSort(arr, left, mid);
            // Sort Right Half
            mergeSort(arr, mid+1, right);
            // Merge Both Halves
            merge(arr, left, mid, right);
        }
    }

    // Merge two sorted subarrays
    public static void merge(int[] arr, int left, int mid, int right){
        int n1 = mid-left+1;
        int n2 = right-mid;
        int[] l = new int[n1];
        int[] r = new int[n2];

        // Copy data to temp arrays
        for(int i=0; i<n1; i++)
            l[i] = arr[left+i];
        for(int j=0; j<n2; j++)
            r[j] = arr[mid+1+j];

        int i = 0, j=0, k=left;

        // Merge temp arrays back into array/arr
        while (i<n1 && j<n2) {
            if (l[i]<=r[j]) {
                arr[k++] = l[i++];
            }
            else{
                arr[k++] = r[j++];
            }
        }

        // Copy remaining elements
        while (i<n1) {
            arr[k++] = l[i++];
        }
        while (j<n2) {
            arr[k++] = r[j++];
        }
    }

    // Driver Code
    public static void main(String[] args) {

        int[] arr = {38,27,43,3,9,82,10};
        System.out.println("Before Sorting: ");
        System.out.println(Arrays.toString(arr));

        mergeSort(arr, 0, arr.length-1);

        System.out.println("After Sorting: ");
        System.out.println(Arrays.toString(arr));

    }
}
```

Output



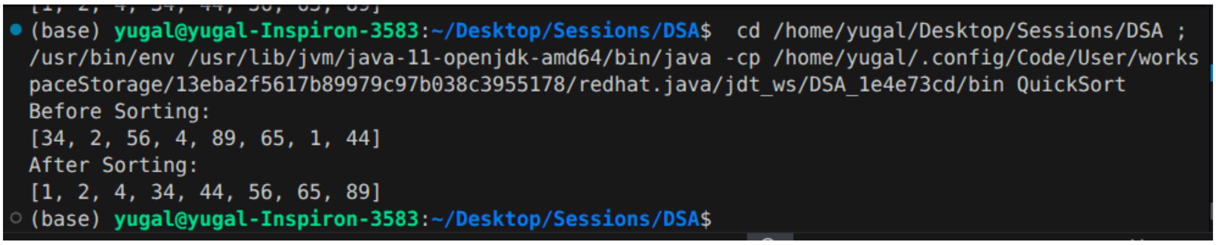
```
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  cd /home/yugal/Desktop/Sessions/DSA ;
/usr/bin/env /usr/lib/jvm/java-11-openjdk-amd64/bin/java -cp /home/yugal/.config/Code/User/works
paceStorage/13eba2f5617b89979c97b038c3955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin MergeSort
Before Sorting:
[38, 27, 43, 3, 9, 82, 10]
After Sorting:
[3, 9, 10, 27, 38, 43, 82]
(base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 6: Program to show Merge Sort

2.5 Write a program to show Quick Sort

```java
    // Quick Sort : small -- pivot -- larger
import java.util.Arrays;

public class QuickSort {

    // Main quick sort function
    public static void quickSort(int[] arr, int low, int high){
        if (low<high) {
            int pivotIndex = partition(arr, low, high);

            // Sort elements before and after partition
            quickSort(arr, low, pivotIndex-1);
            quickSort(arr, pivotIndex+1, high);
        }
    }

    // Partition Function
    public static int partition(int[] arr, int low, int high){
        int pivot = arr[high]; // Choose last element as pivot
        int i = low-1;
        for(int j=low; j<high; j++){
            if (arr[j]<pivot) {
                i++;

                // Swap arr[i] and arr[j]
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }

        // Place pivot at correct position
        int temp = arr[i+1];
        arr[i+1] = arr[high];
        arr[high] = temp;

        return i+1;
    }

    // Driver code
    public static void main(String[] args) {

        int[] arr = {34,2,56,4,89,65,1,44};
        System.out.println("Before Sorting: ");
        System.out.println(Arrays.toString(arr));

        quickSort(arr, 0, arr.length-1);
        System.out.println("After Sorting: ");
        System.out.println(Arrays.toString(arr));

    }
}
```

Output



```
[1, 2, 4, 34, 44, 56, 65, 89]
● (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$  cd /home/yugal/Desktop/Sessions/DSA ;
/usr/bin/env /usr/lib/jvm/java-11-openjdk-amd64/bin/java -cp /home/yugal/.config/Code/User/works
paceStorage/13eba2f5617b89979c97b038c3955178/redhat.java/jdt_ws/DSA_1e4e73cd/bin QuickSort
Before Sorting:
[34, 2, 56, 4, 89, 65, 1, 44]
After Sorting:
[1, 2, 4, 34, 44, 56, 65, 89]
○ (base) yugal@yugal-Inspiron-3583:~/Desktop/Sessions/DSA$
```

Figure 7: Program to show Quick Sort