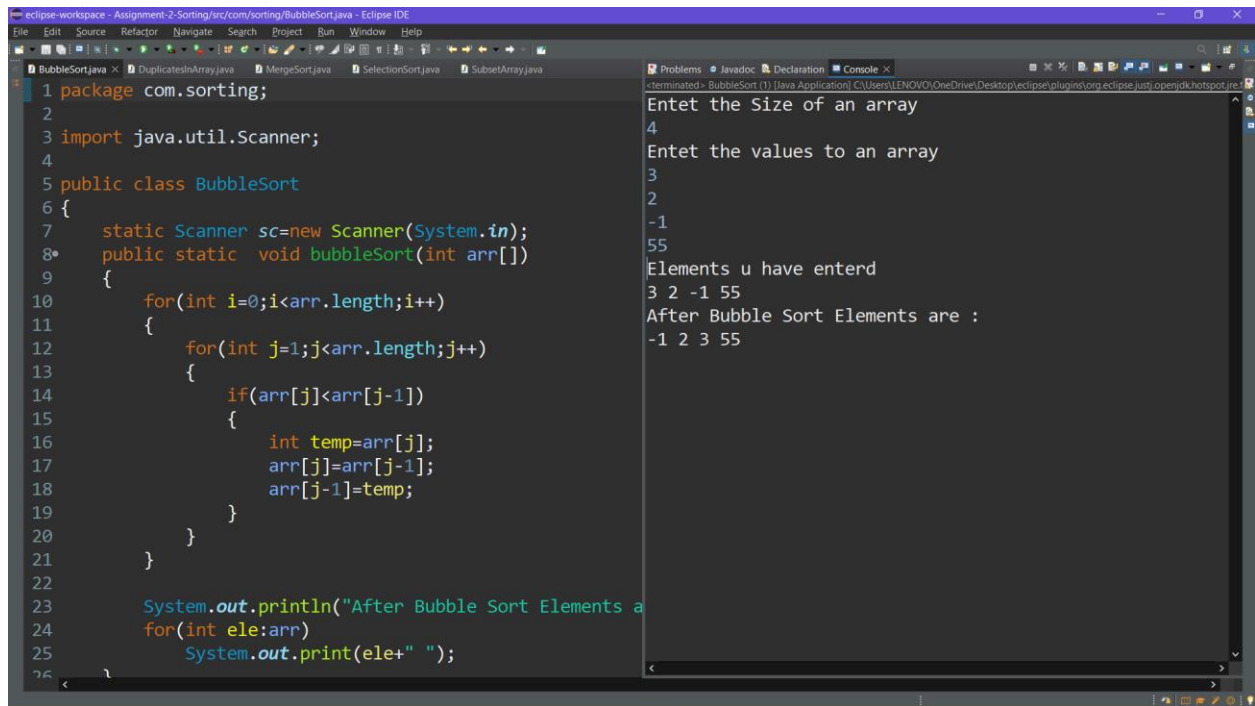


## SORTING OUTPUTS

### BUBBLE SORT



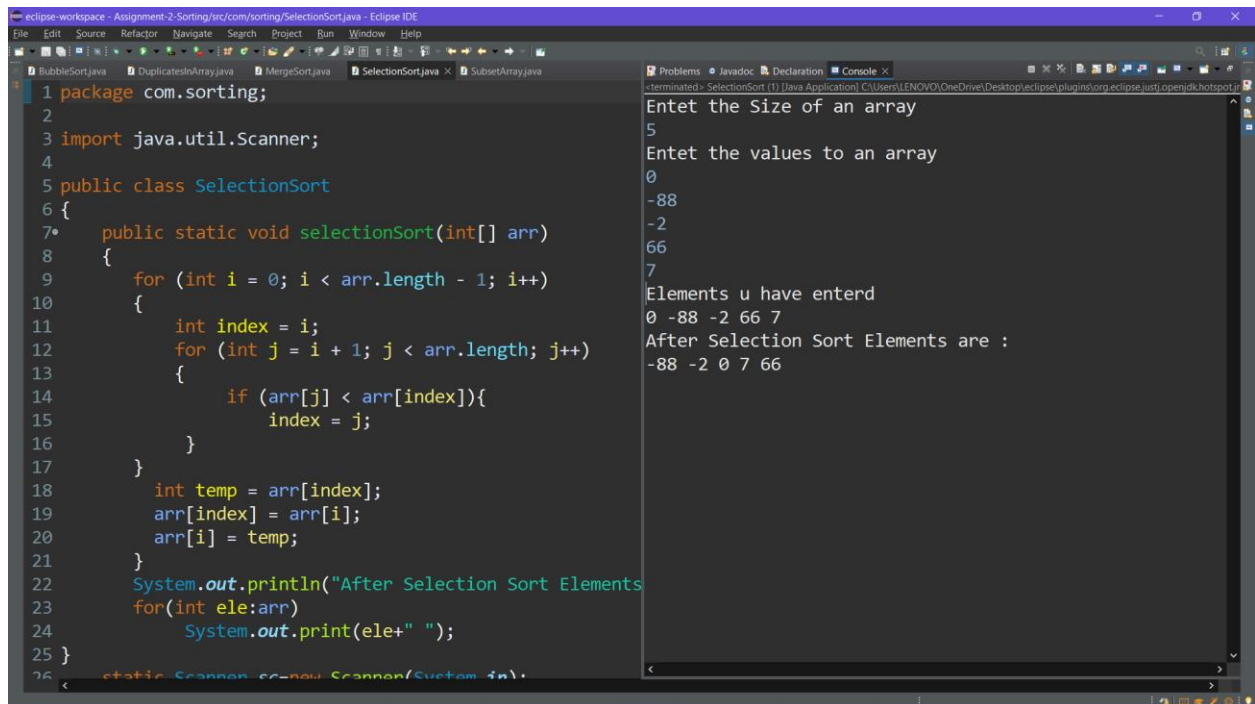
The screenshot shows the Eclipse IDE with the file `BubbleSort.java` open. The code implements a bubble sort algorithm. The console output shows the program's execution, including prompts for array size and values, and the final sorted array.

```
1 package com.sorting;
2
3 import java.util.Scanner;
4
5 public class BubbleSort
6 {
7     static Scanner sc=new Scanner(System.in);
8     public static void bubbleSort(int arr[])
9     {
10         for(int i=0;i<arr.length;i++)
11         {
12             for(int j=1;j<arr.length;j++)
13             {
14                 if(arr[j]<arr[j-1])
15                 {
16                     int temp=arr[j];
17                     arr[j]=arr[j-1];
18                     arr[j-1]=temp;
19                 }
20             }
21         }
22
23         System.out.println("After Bubble Sort Elements are :");
24         for(int ele:arr)
25             System.out.print(ele+" ");
26     }
27 }
```

Console Output:

```
<terminated> BubbleSort (1) [Java Application] C:\Users\LENOVO\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre8.win32.x86_64\jre\bin\java.exe
Entet the Size of an array
4
Entet the values to an array
3
2
-1
55
Elements u have enterd
3 2 -1 55
After Bubble Sort Elements are :
-1 2 3 55
```

### SELECTION SORT:



The screenshot shows the Eclipse IDE with the file `SelectionSort.java` open. The code implements a selection sort algorithm. The console output shows the program's execution, including prompts for array size and values, and the final sorted array.

```
1 package com.sorting;
2
3 import java.util.Scanner;
4
5 public class SelectionSort
6 {
7     public static void selectionSort(int[] arr)
8     {
9         for (int i = 0; i < arr.length - 1; i++)
10         {
11             int index = i;
12             for (int j = i + 1; j < arr.length; j++)
13             {
14                 if (arr[j] < arr[index]){
15                     index = j;
16                 }
17             }
18             int temp = arr[index];
19             arr[index] = arr[i];
20             arr[i] = temp;
21         }
22         System.out.println("After Selection Sort Elements are :");
23         for(int ele:arr)
24             System.out.print(ele+" ");
25     }
26 }
```

Console Output:

```
<terminated> SelectionSort (1) [Java Application] C:\Users\LENOVO\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre8.win32.x86_64\jre\bin\java.exe
Entet the Size of an array
5
Entet the values to an array
0
-88
-2
66
7
Elements u have enterd
0 -88 -2 66 7
After Selection Sort Elements are :
-88 -2 0 7 66
```

### MERGE SORT:

The screenshot shows the Eclipse IDE with the file `MergeSort.java` open. The code implements a merge sort algorithm. The console output shows the program's execution, including prompts for array size and values, and the resulting sorted array.

```
1 package com.sorting;
2
3 import java.util.Scanner;
4
5 public class MergeSort
6 {
7
8     static Scanner sc=new Scanner(System.in);
9     void merge(int arr[], int l, int m, int r)
10    {
11
12        int n1 = m - l + 1;
13        int n2 = r - m; int L[] = new int[n1]; int R[]
14
15        for (int i = 0; i < n1; ++i)
16            L[i] = arr[l + i];
17
18        for (int j = 0; j < n2; ++j)
19            R[j] = arr[m + 1 + j];
20
21        int i = 0, j = 0; int k = l;
22
23        while (i < n1 && j < n2)
24        {
25            if (L[i] <= R[j])
26                arr[k] = L[i]; i++;
27            else
28                arr[k] = R[j]; j++;
29            k++;
30        }
31        while (i < n1)
32            arr[k] = L[i]; i++; k++;
33        while (j < n2)
34            arr[k] = R[j]; j++; k++;
35    }
36}
```

Console Output:

```
<terminated> MergeSort (1) [Java Application] C:\Users\LENOVO\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre16
Entet the Size of an array
6
Entet the values to an array
0
-5
3
2
1
5
Before Sorting
0 -5 3 2 1 5
After Merge Sort array
-5 0 1 2 3 5
```

Find Duplicates:

The screenshot shows the Eclipse IDE with the file `DuplicatesInArray.java` open. The code finds duplicate elements in an array using an ArrayList. The console output shows the program's execution, including prompts for array size and values, and the identified duplicates.

```
1 package com.sorting;
2
3 import java.util.ArrayList;
4
5 public class DuplicatesInArray
6 {
7
8
9     static Scanner sc=new Scanner(System.in);
10    public static void findDuplicates(int arr[])
11    {
12
13        ArrayList<Integer>ar=new ArrayList<Integer>();
14
15        for(int i=0;i<arr.length;i++)
16        {
17            for(int j=i+1;j<arr.length;j++)
18            {
19                if(arr[i]==arr[j])
20                {
21                    ar.add(arr[i]);
22                }
23            }
24        }
25        System.out.println("Duplicates in above array");
26        for(int ele:ar)
27            System.out.print(ele+" ");
28    }
29}
```

Console Output:

```
<terminated> DuplicatesInArray (1) [Java Application] C:\Users\LENOVO\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre16
Entet the Size of an array
6
Entet the values to an array
2
2
3
3
4
5
Duplicates in above array
2 3
```

Check SubsetArray:

```
eclipse-workspace - Assignment-2-Sorting/src/com/sorting/SubsetArray.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BubbleSort.java DuplicatesInArray.java MergeSort.java SelectionSort.java SubsetArray.java
1 package com.sorting;
2
3 import java.util.Scanner;
4
5 public class SubsetArray
6 {
7     static Scanner sc=new Scanner(System.in);
8     public static void subsetArray(int arr[],int arr1[])
9     {
10         if(arr.length>arr1.length)
11         {
12             int c=0;
13             for(int i=0;i<arr.length;i++)
14             {
15                 for(int j=0;j<arr1.length;j++)
16                 {
17                     if(arr[i]==arr1[j])
18                         c=c+1;
19                 }
20             }
21             if(arr1.length==c)
22                 System.out.println("Array2 is Subset of
23             else
24                 System.out.println("Array2 is not Subset
25         }
26     }
27 }
```

```
Problems Javadoc Declaration Console x
<terminated> SubsetArray (1) [Java Application] C:\Users\LENOVO\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre
Entet the Size of an array 1
4
Entet the values to an array 1
1
2
3
4
Entet the Size of an array 2
2
Entet the values to an array 2
3
4
Array2 is Subset of Array 1 :)
```

```
eclipse-workspace - Assignment-2-Sorting/src/com/sorting/SubsetArray.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BubbleSort.java DuplicatesInArray.java MergeSort.java SelectionSort.java SubsetArray.java
1 package com.sorting;
2
3 import java.util.Scanner;
4
5 public class SubsetArray
6 {
7     static Scanner sc=new Scanner(System.in);
8     public static void subsetArray(int arr[],int arr1[])
9     {
10         if(arr.length>arr1.length)
11         {
12             int c=0;
13             for(int i=0;i<arr.length;i++)
14             {
15                 for(int j=0;j<arr1.length;j++)
16                 {
17                     if(arr[i]==arr1[j])
18                         c=c+1;
19                 }
20             }
21             if(arr1.length==c)
22                 System.out.println("Array2 is Subset of
23             else
24                 System.out.println("Array2 is not Subset
25         }
26     }
27 }
```

```
Problems Javadoc Declaration Console x
<terminated> SubsetArray (1) [Java Application] C:\Users\LENOVO\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre
Entet the Size of an array 1
3
Entet the values to an array 1
1
2
3
Entet the Size of an array 2
3
Entet the values to an array 2
4
5
6
Array1 is not Subset of Array 2 :)
```

The screenshot shows the Eclipse IDE with a Java project named 'Assignment-2-Sorting'. The editor displays the file 'SubsetArray.java' with the following code:

```
1 package com.sorting;
2
3 import java.util.Scanner;
4
5 public class SubsetArray
6 {
7     static Scanner sc=new Scanner(System.in);
8     public static void subsetArray(int arr[],int arr1[])
9     {
10         if(arr.length>arr1.length)
11         {
12             int c=0;
13             for(int i=0;i<arr.length;i++)
14             {
15                 for(int j=0;j<arr1.length;j++)
16                 {
17                     if(arr[i]==arr1[j])
18                         c=c+1;
19                 }
20             }
21             if(arr1.length==c)
22                 System.out.println("Array2 is Subset of
23             else
24                 System.out.println("Array2 is not Subset
25         }
26     }
```

The console output on the right shows the program's execution:

```
Enter the Size of an array 1
2
Enter the values to an array 1
1
2
Enter the Size of an array 2
4
Enter the values to an array 2
1
2
3
4
Array1 is Subset of Array 2 ):
```

Quick Sort: