

Experiment No._____

Date____/____/2020

TITLE OF EXPERIMENT: - A program in Java to Sort

DIVISION:_____ **BRANCH:** _____

BATCH:_____ **ROLL NO.:** _____

PERFORMED ON DATE: _____

SIGNATURE OF TEACHING STAFF:

EXPERIMENT NO. 5

Aim: Write Programs in Java to sort i) List of integers ii) List of names.

Objective: To learn Arrays and Strings in Java

Software:

1. Eclipse
2. JDK 16

Theory:

i) Java Program to sort the names of an array in ascending order

For, sorting names in an Alphabetical order there are multiple ways to sort the array, like using inbuilt Arrays.sort() method or using normal sorting algorithms like the bubble sort, merge sort. Here let's use the bubble sort and inbuilt sort.

Example:

Input : Array[] = {"Sourabh", "Anoop", "Harsh", "Alok", "Tanuj"}

Output: Array[] = {"Alok", "Anoop", "Harsh", "Sourabh", "Tanuj"}

Input : Array[] = {"Bob", "Alice"}

Output: Array[] = {"Alice", "Bob"}

1. Brute-Force Approach

The idea is to compare the strings on the basis of their unicode and swap them in accordance with the returned int value based on the comparison between the two strings using compareTo() method.

In the input, the user has to enter the number of names and the names and on the output, it will sort and display them in alphabetical order. For this, we are going to use the compareTo() method and compare one string with the rest of the strings.

CompareTo() is used to compare two strings lexicographically. Each character of both strings is converted into its **unicode** value. Lexicographical order is nothing but alphabetical order.

This method returns an int data-type which is based on the comparison between the two string. **If it returns >0 then the parameter passed to compareTo() method is lexicographically first whereas if returns < 0 then string calling the method is lexicographically correct.**

Steps:

- Using **CompareTo()** method compare one string with the rest of the strings
- To swap the elements based on the comparison between the two string.
- Print the Sorted Names in an Alphabetical Order.

Program:

```
// Java Program to Sort Names in an Alphabetical Order
class sorting {
    public static void main(String[] args)
    {
        // storing input in variable
        int n = 4;

        // create string array called names
        String names[]
            = { "Rahul", "Ajay", "Gourav", "Riya" };
        String temp;
        for (int i = 0; i < n; i++) {
            for (int j = i + 1; j < n; j++) {

                // to compare one string with other strings
                if (names[i].compareTo(names[j]) > 0) {
                    // swapping
                    temp = names[i];
                    names[i] = names[j];
                    names[j] = temp;
                }
            }
        }

        // print output array
        System.out.println(
            "The names in alphabetical order are: ");
        for (int i = 0; i < n; i++) {
            System.out.println(names[i]);
        }
    }
}
```

```
}  
}
```

Output:

The names in alphabetical order are:

Ajay

Gourav

Rahul

Riya

2. Inbuilt Sort function

- Using inbuilt [Arrays.sort\(\)](#) method to sort the array.
- Print the Sorted Names in an Alphabetical Order.

Below is the implementation of the above approach:

Java

```
// Java Program to Sort Names in an Alphabetical Order by using inbuilt sort  
function  
  
import java.io.*;  
  
import java.util.*;  
  
class GFG {  
  
    public static void main(String[] args)  
  
    {  
  
        // storing input in variable  
  
        int n = 4;  
  
        // create string array called names  
  
        String names[]
```

```
        = { "Rahul", "Ajay", "Gourav", "Riya" };

// inbuilt sort function

Arrays.sort(names);

// print output array

System.out.println(

    "The names in alphabetical order are: ");

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

        System.out.println(names[i]);

    }

}

}
```

Output

The names in alphabetical order are:

Ajay

Gourav

Rahul

Riya

ii) Java Program to sort the elements of an array in ascending order

In this program, we need to sort the given array in ascending order such that elements will be arranged from smallest to largest. This can be achieved through two loops. The outer loop will select an element, and inner loop allows us to compare selected element with rest of the elements.

Original array:

5 2 8 7 1

Array after sorting:

1 2 5 7 8

Elements will be sorted in such a way that the smallest element will appear on extreme left which in this case is 1. The largest element will appear on extreme right which in this case is 8.

Algorithm

- **STEP 1:** START
- **STEP 2:** INITIALIZE arr[] = {5, 2, 8, 7, 1 }.
- **STEP 3:** SET temp =0
- **STEP 4:** PRINT "Elements of Original Array"
- **STEP 5:** REPEAT STEP 6 UNTIL i<arr.length
 //for(i=0; i<arr.length; i++)
- **STEP 6:** PRINT arr[i]
- **STEP 7:** REPEAT STEP 8 to STEP 9 UNTIL i<arr.length
 //for(i=0; i<arr.length; i++)
- **STEP 8:** REPEAT STEP 9 UNTIL j<arr.length
 //for(j=i+1;j<arr.length;j++)
- **STEP 9:** if(arr[i]>arr[j]) then
 temp = arr[i]
 arr[i]=arr[j]
 arr[j]=temp
- **STEP 10:** PRINT new line

- **STEP 11:** PRINT "Elements of array sorted in ascending order"
- **STEP 12:** REPEAT STEP 13 UNTIL i<arr.length
//for(i=0;i<arr.length;i++)
- **STEP 13:** PRINT arr[i]
- **STEP 14:** END

Program:

// Java Program to Sort integer in an Ascending Order

```
public class SortAsc {  
    public static void main(String[] args) {  
  
        //Initialize array  
        int [] arr = new int [] {5, 2, 8, 7, 1};  
        int temp = 0;  
  
        //Displaying elements of original array  
        System.out.println("Elements of original array: ");  
        for (int i = 0; i < arr.length; i++) {  
            System.out.print(arr[i] + " ");  
        }  
  
        //Sort the array in ascending order  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = i+1; j < arr.length; j++) {  
                if(arr[i] > arr[j]) {  
                    temp = arr[i];  
                    arr[i] = arr[j];  
                    arr[j] = temp;  
                }  
            }  
        }  
  
        System.out.println();  
    }  
}
```

```
//Displaying elements of array after sorting
System.out.println("Elements of array sorted in ascending order: ");
for (int i = 0; i < arr.length; i++) {
    System.out.print(arr[i] + " ");
}
}
```

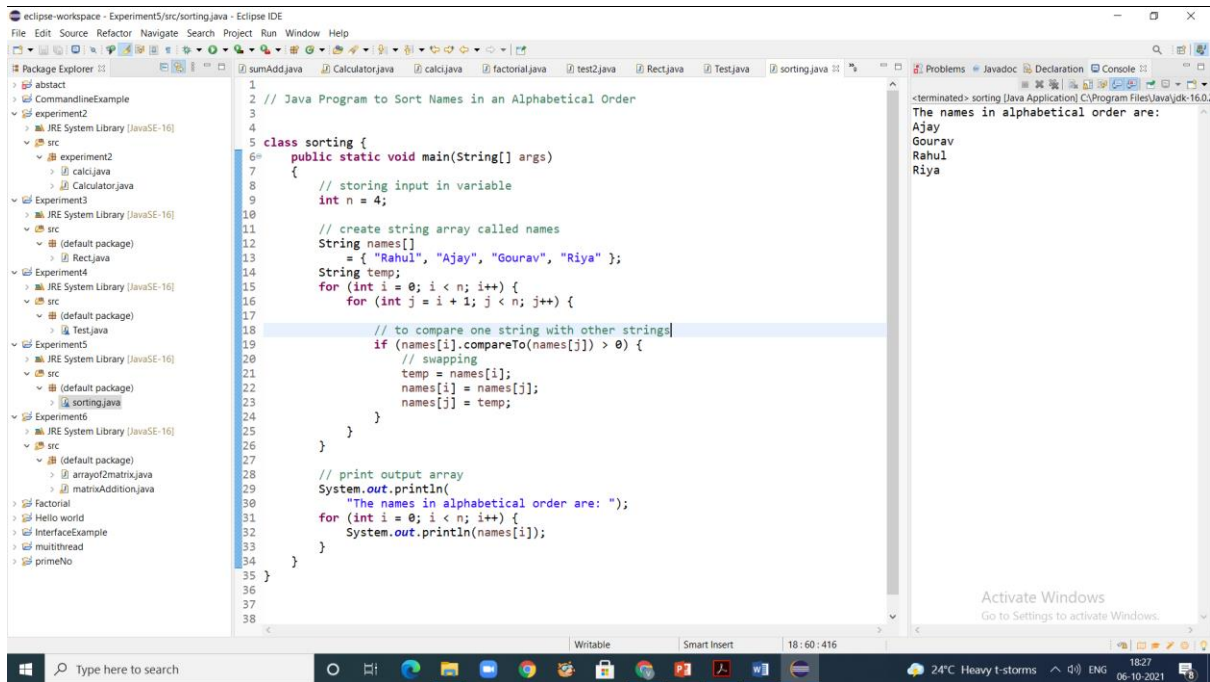
Output:

```
Elements of original array:
5 2 8 7 1
Elements of array sorted in ascending order:
1 2 5 7 8
```

Conclusion:

Screenshot's of Program and Result:

i) Java Program to Sort Names in an Alphabetical Order



```
1 // Java Program to Sort Names in an Alphabetical Order
2
3
4
5 class sorting {
6     public static void main(String[] args)
7     {
8         // storing input in variable
9         int n = 4;
10
11         // create string array called names
12         String names[]
13             = { "Rahul", "Ajay", "Gourav", "Riya" };
14         String temp;
15         for (int i = 0; i < n; i++) {
16             for (int j = i + 1; j < n; j++) {
17
18                 // to compare one string with other strings
19                 if (names[i].compareTo(names[j]) > 0) {
20                     // swapping
21                     temp = names[i];
22                     names[i] = names[j];
23                     names[j] = temp;
24                 }
25             }
26         }
27
28         // print output array
29         System.out.println(
30             "The names in alphabetical order are: ");
31         for (int i = 0; i < n; i++) {
32             System.out.println(names[i]);
33         }
34     }
35 }
36
37
38
```

The console output shows the names sorted in alphabetical order: Ajay, Gourav, Rahul, Riya.

ii) Java Program to Sort integer in Ascending Order

