

Lab 08

One Dimensional Arrays

Objective:

In this lab, students will know about the basic concepts of Array data structure, storing different types of data in arrays, creating arrays of integers, double, Strings and passing arrays to the methods.

Activity Outcomes:

The activities provide hands - on practice with the following topics

- Creating arrays
- Accessing indexes/location of variables in arrays
- Accessing maximum, minimum numbers in arrays
- Different array's operation
- Passing arrays to methods

Instructor Note:

As pre-lab activity, read Chapter 7 from the text book “Java How to Program, Deitel, P. & Deitel, H., Prentice Hall, 2019”.

1) Useful Concepts

Often you will have to store a large number of values during the execution of a program. Suppose, for instance, that you need to read 100 numbers, compute their average, and find out how many numbers are above the average. Your program first reads the numbers and computes their average, then compares each number with the average to determine whether it is above the average. In order to accomplish this task, the numbers must all be stored in variables. You have to declare 100 variables and repeatedly write almost identical code 100 times. Writing a program this way would be impractical. So, how do you solve this problem?

An efficient, organized approach is needed. Java and most other high-level languages provide a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. In the present case, you can store all 100 numbers into an array and access them through a single array variable.

Array is a data structure that represents a collection of the same types of data.

The array elements are accessed through the index. The array indices are 0-based, i.e., it starts from 0 to `arrayRefVar.length-1`.

1. **Declaring arrays:** An array can be declared as below.

```
elementType[] arrayRefVar;
```

The elementType can be any data type, and all elements in the array will have the same data type. For example, the following code declares a variable myList that references an array of double elements.

```
double[] myArray;
```

2. **Creating arrays:** Once array is declared, it can be created as below.

```
arrayRefVar = new elementType[arraySize];
```

In our case it can be as below.

```
myArray = new double[10];
```

3. We can also define arrays as below.

```
double[] myArray = new double[10];
```

2) Solved Lab Activities

Sr.No	Allocated Time	Level of Complexity	CLO Mapping
Activity 1	15 mins	Midum	CLO-5
Activity 2	15 mins	Midum	CLO-5
Activity 3	15 mins	Midum	CLO-5
Activity 4	15 mins	Midum	CLO-5

Activity 1:

Write a Java program to accept an array of 10 integer values from user and find the largest and second largest values.

Solution:

```
1 import java.util.Scanner;
2 public class ArrayExample
3 {
4     public static void main(String args[])
5     {
6         Scanner input=new Scanner(System.in);
7         int[] array=new int[10];
8         System.out.println("Enter array elements...")
9
10        for(int i=0;i<10;i++)
11        {
12            array[i]=input.nextInt();
13        }
14        int largest=array[0];
15        for(int i=0;i<10;i++)
16        {
17            if(array[i]>largest)
18                largest=array[i];
19        }
20        int largest2;
21        if(array[0]==largest)
22            largest2=array[1];
23        else
24            largest2=array[0];
25        for(int i=0;i<10;i++)
26        {
27            if(array[i]!=largest)
28                if(array[i]>largest2)
29                    largest2=array[i];
30        }
31        System.out.println("Largest="+largest);
32        System.out.println("2nd Largest="+largest2);
33    }
}
```

Output

```
| Enter array elements...
```

```
| 9
```

```
| 1
```

```
| 22
```

```
| 11
```

```
| 99
```

```
| 2
```

```
| 8
```

```
| 3
```

```
| 7
```

```
| 6
```

```
| Largest=99
```

```
| 2nd Largest=22
```

```
| Process completed.
```

Activity 2:

Write a Java program to accept 5 integer values from user. Pass this array to a method to find the sum of the array.

Solution:

```
1 //Program to find sum of the given integer array
2 import java.util.Scanner;
3 class SumArray
4 {
5     public static void main(String args[])
6     {
7         Scanner input=new Scanner(System.in);
8         int[] array=new int[5];
9         System.out.println("Enter array elements...");
10        for(int i=0;i<array.length;i++)
11        {
12            array[i]=input.nextInt();
13        }
14        System.out.println("Sum of the given array is... "+sum(array));
15    }
16    static int sum(int[] arr)
17    {
18        int s=0;
19        for(int x:arr)
20            s+=x;
21        return s;
22    }
23 }
```

Output:

```
Enter array elements...  
1  
2  
3  
4  
5  
Sum of the given array is... 15  
Process completed.
```

Activity 3:

Write a program which takes 50 characters as input in array and counts the occurrences of each character.

E.g. A occurs 2 times
Y occurs 1 time
E occurs 1 time
O occurs 2 times
K occurs 1 time
@ occurs 1 time
..... So on...

Solution:

```

1 import java.util.Scanner;
2
3 public class numberofoccurrences {
4
5     public static void main(String[] args) {
6         char[] characters = new char[10];
7         int size = 0;
8         int element = characters[0];
9         char[] unique = new char[size];
10        int counter = 0;
11        Scanner input = new Scanner(System.in);
12
13        //////Input in array//////
14        for (int i = 0; i < 10; i++) {
15            System.out.print("A[" + i + "] = ");
16            characters[i] = input.next().charAt(0);
17        }
18        for (int i = 0; i < 10; i++) {
19            for (int j = 0; j < 10; j++) {
20
21                if (characters[i] == characters[j]) {
22                    counter = counter + 1;
23                }
24            }
25            System.out.println(characters[i] + " occurs " + counter + " times");
26            counter = 0;
27        }
28    }
29}

```

Output

```

run:
A[0] = 5
A[1] = 7
A[2] = 4
A[3] = 2
A[4] = 1
A[5] = 4
A[6] = 6
A[7] = 8
A[8] = 9
A[9] = 5
5 occurs 2 times
7 occurs 1 times
4 occurs 2 times
2 occurs 1 times
1 occurs 1 times
4 occurs 2 times
6 occurs 1 times
8 occurs 1 times
9 occurs 1 times
5 occurs 2 times
BUILD SUCCESSFUL (total time: 24 seconds)

```

Activity 4:

Create a menu driven program, with the following functionalities: (Note: all of the functionalities should be created in a single program with following menu options)

1. Input elements in array. (details of this functionality is given in Step a)
2. Search element and its location. (details of this functionality is given in Step b)
3. Find largest & smallest value in the array. (details of this functionality is given in Step c)
4. Copy data. (details of this functionality is given in Step d)
 - a) Input 10 elements in the array and display the array. (Note: this should call two methods `Input(int Array[])` and `display(int A[])`)
 - b) Search element is in the array then print "Element found" along with its location. (Note: this should call two methods `Input(int Array[])` and `search(int searchkey, int Array[])`. You should call the same `Input()` method that is called in step a)
 - c) Find the largest and the smallest element in array. Then place largest element on 0th index and smallest element on the last index 9th. (Note: this should call three methods previously used `Input(int Array[])` , `Largest(int Array[])` and `Smallest (int Array[])`)
 - d) Copy the contents of one array into another.(Note: this should call two methods `Input(int Array[])` and `copydata(int Array[], int copiedArray[])` .

Solution:

```
1  import java.util.Scanner;
2
3  public class ArraysImplementation {
4      // Main Method
5
6      public static void main(String[] args) {
7          // Menu Option place
8          Scanner input = new Scanner(System.in);
9          System.out.println(" Enter a number to choose menu : ");
10         System.out.println(" 1 : For input and display the Array \n 2 : For searching Element in an Array ");
11         System.out.println(" 3 : For largest and Smallest element in the Array ");
12         System.out.println(" 4 : For copy data to another Array");
13
14         int a = input.nextInt();
15
16         // Condition checking
17         switch (a) {
18             // For input and Display
19             case 1: {
20                 int[] a1 = new int[1];
21                 int[] array = (Input(a1));
22                 display(array);
23                 break;
24             }
25             case 2: {
26                 int[] a1 = new int[1];
27                 int[] array1 = (Input(a1));
28                 System.out.println(" Enter number for search");
29                 int searchkey = input.nextInt();
30                 search(array1, searchkey);
31                 break;
32             }
33         }
34     }
35 }
```

```

35 // For Largest and Smallest in the Array
36     case 3: {
37         int[] a1 = new int[1];
38         int[] array11 = (Input(a1));
39         largest(array11);
40         smallest(array11);
41         break;
42     }
43 // For Copying to new Array
44     case 4: {
45         int[] a1 = {1};
46         int[] array1 = (Input(a1));
47         int[] copied = new int[10];
48         copydata(array1, copied);
49         break;
50     }
51     default:
52         System.out.println(" You Enter invalid number");
53     }
54 }
55 // Input array Method
56
57 public static int[] Input(int Array[]) {
58     int[] array1 = new int[10];
59     Scanner input = new Scanner(System.in);
60     for (int i = 0; i < 10; i++) {
61         System.out.println(" A[" + i + "] = ");
62         array1[i] = input.nextInt();
63     }
64     return array1;
65 }
66 // Display array Method
67
68 public static void display(int[] A) {
69     for (int j = 0; j < 10; j++) {
70         System.out.println(" A[" + j + "] " + A[j]);
71     }
72 }
73
74 // Searching element in Array Method
75 public static void search(int[] array, int x) {
76     int flag = 0;
77
78     while (flag == 0) {
79         for (int j = 0; j < 10; j++) {
80             if (array[j] == x) {
81                 System.out.println(" Element Found A[" + j + "] " + array[j]);
82                 flag++;
83             }
84         }
85         if (flag == 0) {
86             System.out.println(" Element is not found ");
87             break;
88         }
89     }
90 }
91

```

```

91 // Finding largest element in array Method
92 public static void largest(int[] array) {
93     int k = array[0];
94     for (int i = 0; i < 10; i++) {
95         if (array[i] >= k) {
96             k = array[i];
97         }
98     }
99     System.out.println("Largest value at A[0] " + k);
100 }
101 // Finding Smallest element in array Method
102
103 public static void smallest(int[] array) {
104     int k = array[9];
105     for (int i = 0; i < 10; i++) {
106         if (k >= array[i]) {
107             k = array[i];
108         }
109     }
110     System.out.println("Smallest value at A[9] " + k);
111 }
112 // Copying data to new array Mehtod
113
114 public static void copydata(int array[], int copied[]) {
115     for (int j = 0; j < 10; j++) {
116         copied[j] = array[j];
117     }
118     for (int k = 0; k < 10; k++) {
119         System.out.println(" New copied array is B[" + k + "]" + copied[k]);
120     }
121 }
122 }
123
124 }

```

Output

```

run:
A[0] = 5
A[1] = 7
A[2] = 4
A[3] = 2
A[4] = 1
A[5] = 4
A[6] = 6
A[7] = 8
A[8] = 9
A[9] = 5
5 occurs 2 times
7 occurs 1 times
4 occurs 2 times
2 occurs 1 times
1 occurs 1 times
4 occurs 2 times
6 occurs 1 times
8 occurs 1 times
9 occurs 1 times
5 occurs 2 times
BUILD SUCCESSFUL (total time: 24 seconds)

```

3) Graded Lab Tasks

Note: The instructor can design graded lab activities according to the level of difficult and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab.

Lab Task 1

10 students were asked to rate the quality of food in the student cafeteria, on a scale of 1 to 10 (1 means awful and 10 means excellent). Place the forty responses in an integer array and summarize the results of the poll.

Lab Task 2

Write a program which performs the following tasks:

- Initialize an integer array of 10 elements in main()*
- Pass the entire array to a function modify()*
- In modify() multiply each element of array by 3*
- return the control to main() and print the new array elements in main()*

Lab Task 3

Write a program to copy the contents of one array into another in the reverse order.