Arrays

Module 8 CISS-110

Muhammad Rashid Bhatti

Data Structures:

Data structures are a specific way to store and organize data in different formats. For example, if we have to manually organize phone book on a sheet of paper, we may sort it in alphabetical order of name or sort it on phone number numeric code so that contacts from same area are in one place. The same logic applies to computer programming. We have different data structures to store information. The choice of correct data structure is very important as it can significantly reduce program complexity.

Arrays

An array is used to store a collection of same type variables. For example, we may declare an array of integers that will have many integers stored with one array variable name. Arrays are one of many data structures that are frequently used in software industry to store collection of data to solve real life problems. For example, a programmer can declare an array of objects to store telephone directory. The array can be sorted on people name or phone number to quickly search for a particular name or phone number.

Array programming is an example of Data Structure programming. You will learn data structure programming in advanced programming classes. For this course, we shall study only one data structure Arrays and their use.

Why arrays are used?

If we are required to declare first four odd integers, one way of doing it is:

```c
int firstInt = 1;
int secondInt = 3;
int thirdInt = 5;
int fourthInt = 7;
```
The same could be accomplished by an array, which takes less line of code.

```java
int[] oddArray = {1, 3, 5, 7};
```

**Other way of defining an array:**

Another way of declaring an array is to declare an array object first and then assign values to each element of an array one by one.

```java
int[] oddArray = new int[4];
oddArray[0]=1;
oddArray[1]=3;
oddArray[2]=5;
oddArray[3]=7;
```

Both the declarations are doing the same thing.

**Array indexes:**

The array elements are referenced by their indexes. Array indexes start from 0. In above example, if we have to reference 1, then we shall use `oddArray[0]`. To reference number 7 we shall use `oddArray[3]`. Now, if we want to change the value of 7 to 11 then we shall use indexes. The following command shall change the value of 7 to 11 in above array.

```java
oddArray[3]=11;
```

**Practice programs for array declaration and assigning values:**

Program 8.1 is a good example of how an array is declared. We are also assigning values to each element of an array. In the end we are displaying the result on screen.

```
Program 8.1:

```java
import java.util.Scanner;

// This program takes input 4 times to assign array values at index 0, 1, 2 and 3.
public class ArrayInput {
    static Scanner kb = new Scanner(System.in);
    public static void main(String[] args) {
        // declare an input array of size 4
        int[] inputArray = new int[4];
        int x;
        int len = 4;
        // take input and read data into array using for loop.
```
for (x = 0; x < len; x++) {
    System.out.print("Enter value for array[" + x + "]: ");
    inputArray[x] = kb.nextInt();
}

// call to user defined method printArray to display output
printArray(inputArray, "inputArray");

public static void printArray(int[] a, String type) {
    int x, len = a.length;
    for (x = 0; x < len; x++) {
        // printf is a way to present formatted output. First %d
        // will tell Java that output is integer which is variable
        // x, %s will tell that output will be a string variable
        // 'type' and then last %d will tell that output is
        // integer which is array content represented by
        // variable a[x]
        System.out.printf("Value at index %d for %s is %d", x, type, a[x]);
        System.out.println(" ");
    }
    System.out.println(" ");
}

Output:
Enter value for array[0]: 2
Enter value for array[1]: 4
Enter value for array[2]: 5
Enter value for array[3]: 9
Value at index 0 for inputArray is 2
Value at index 1 for inputArray is 4
Value at index 2 for inputArray is 5
Value at index 3 for inputArray is 9

Practice Program 8.2 is another example of exploring different ways to declare and assign values to an array. There are two arrays, evenArray and oddArray.

- The program declares and assigns values to oddArray at same time.
- The program declares even array first and then assign values to it one by one. Finally both arrays are displayed on output through user defined printArray method.
Program 8.2

import java.util.Scanner;

// This program is an example to show different ways by which we can declare and assign values to an array
public class ArrayDeclaration {
    static Scanner kb = new Scanner(System.in);
    public static void main(String[] args) {
        // The array declaration and assigning values are done
        // in one command
        int[] oddArray = {1, 3, 5, 7};
        // The array has been declared with size 4
        int[] evenArray = new int[4];
        // assign data value for even array element
        evenArray[0]=2;
        evenArray[1]=4;
        evenArray[2]=6;
        evenArray[3]=8;
        // call to user defined method printArray to display
        // output for oddArray
        printArray(oddArray, "oddArray");
        // call to user defined method printArray to display
        // output for evenArray
        printArray(evenArray, "evenArray");
    }
    public static void printArray(int[] a, String type) {
        int x, len = a.length;
        for (x = 0; x < len; x++)
        {
            // printf is a way to present formatted output. First %d
            // will tell Java that output is integer which is variable
            // x, %s will tell that output will be a string variable
            // 'type' and then last %d will tell that output is
            // integer which is array content represented by
            // variable a[x]
            System.out.printf("Value at index %d for %s is %d", x, type, a[x]);
        }
    }
}

Output:

Value at index 0 for oddArray is 1
Value at index 1 for oddArray is 3
Value at index 2 for oddArray is 5
Value at index 3 for oddArray is 7

Value at index 0 for evenArray is 2
Value at index 1 for evenArray is 4
Value at index 2 for evenArray is 6
Value at index 3 for evenArray is 8

**Array processing example:**

Following example show how we can add values of all contents of an array. The program also shows technique to find minimum value element in an array. Both of the task has been carried out by user defined methods `sumArray` and `minArray` that takes the array as an input.

Program 8.3

```java
public class ArrayProcess {
    public static void main(String[] args) {
        double[] myArray = {2.1, 3.2, 4.5};

        // Print all the array elements
        for (int i = 0; i < myArray.length; i++) {
            System.out.println(myArray[i] + " ");
        }

        // Summing all elements of array
        System.out.println("Total is "+ sumArray(myArray));

        // Minimum of array is found in minArray user defined method
        System.out.println("Minimum is "+ minArray(myArray));
    }

    public static double sumArray(double[] myArray) {
        // initialize min element value to first element of array
        double total = 0;
        // scan the whole array elements by for loop one by one and
        // then add them to total variable
        for (int i = 0; i < myArray.length; i++) {
            total = total + myArray[i];
        }
        return total;
    }

    public static double minArray(double[] myArray) {
        // initialize min element value to first element of array
        double min = myArray[0];
        // scan the whole array elements by for loop one by one
        // compare if the array element is less than min element
    }
```
Arrays are used for many purposes. Arrays are used to quickly find an element within array. Java JDK uses index number to quickly reach a particular element of array. Another use of array is to sort same data elements. For example, we can store an organization’s employee’s salary in an array. Sorting this array will help us to know which employee has the minimum and maximum salary.

Program 8.4 is a good example of how array sort could be achieved.

```
import java.util.Arrays;
import java.util.Scanner;

// This program takes input 4 times to assign array values at array
// index 0, 1, 2 and 3 and then uses Java pre defined method call sort
// to sort the array.
public class ArraySort {
    static Scanner kb = new Scanner(System.in);
    public static void main(String[] args) {
        // declare an input array of size 4
        int[] inputArray = new int[4];
        int x;
        int len = 4;
        // take input and read data into array using for loop.
        for (x = 0; x < len; x++) {
            System.out.print("Enter value for array[" + x + "]: ");
```
```java
inputArray[x] = kb.nextInt();
}
// call to Java sort command to sort an array. This will // sort in ascending order
Arrays.sort(inputArray);
printArray(inputArray, "inputArray");
}

public static void printArray(int[] a, String type)
{
    int x, len = a.length;
    for (x = 0; x < len; x++) {
        System.out.printf("Value at index %d for %s is %d", x ,
        type, a[x]);
        System.out.println(" ");
    }
    System.out.println(" ");
}

Output:

Enter value for array[0]: 1
Enter value for array[1]: 4
Enter value for array[2]: 3
Enter value for array[3]: 2
Value at index 0 for inputArray is 1
Value at index 1 for inputArray is 2
Value at index 2 for inputArray is 3
Value at index 3 for inputArray is 4