SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai
Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &
Accredited by NBA (B.E - CSE, EEE, ECE, Mech & B.Tech.IT)
COIMBATORE-641 035, TAMIL NADU

UNIT II – Control Statements and Constructors

Control structures – **Arrays** - Objects and classes: Classes – Access Specifiers – methods and attributes - constructors: Default Constructor – Parameterized Constructor – Copy Constructor- Garbage collection.

Arrays

An array is a collection of similar type of elements which has contiguous memory location.

Java array is an object which contains elements of a similar data type. Additionally, the elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

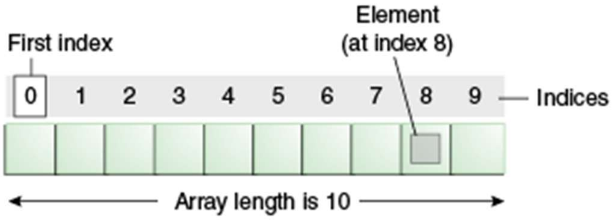
Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

Unlike C/C++, we can get the length of the array using the length member. In C/C++, we need to use the sizeof operator.

In Java, array is an object of a dynamically generated class. Java array inherits the Object class, and implements the Serializable as well as Cloneable interfaces.

We can store primitive values or objects in an array in Java. Like C/C++, we can also create single dimensional or multidimensional arrays in Java.

Moreover, Java provides the feature of anonymous arrays which is not available in C/C++.



Advantages

- Code Optimization: It makes the code optimized, we can retrieve or sort the data efficiently.
- Random access: We can get any data located at an index position.

Disadvantages

- Size Limit: We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.

Types

There are two types of array.
Single Dimensional Array
Multidimensional Array

Single Dimensional Array in Java

Syntax to Declare an Array

```
dataType[] arr; (or)
dataType []arr; (or)
dataType arr[];
```

Instantiation of an Array in Java

```
arrayRefVar=new datatype[size];
```

Example

```
//Java Program to illustrate how to declare, instantiate, initialize
//and traverse the Java array.
class Testarray{
public static void main(String args[]){
int a[]=new int[5];//declaration and instantiation
a[0]=10;//initialization
a[1]=20;
a[2]=70;
a[3]=40;
a[4]=50;
//traversing array
for(int i=0;i<a.length;i++)//length is the property of array
System.out.println(a[i]);
}}
```

Output:

```
10
20
70
40
50
```

Multidimensional Array

data is stored in row and column based index (also known as matrix form).

Syntax to Declare Multidimensional Array

```
dataType[][] arrayRefVar; (or)
dataType [][]arrayRefVar; (or)
dataType arrayRefVar[][]; (or)
dataType []arrayRefVar[];
```

Example to instantiate Multidimensional Array

```
int[][] arr=new int[3][3];//3 row and 3 column
```

Example to initialize Multidimensional Array

```
arr[0][0]=1;
arr[0][1]=2;
arr[0][2]=3;
arr[1][0]=4;
arr[1][1]=5;
arr[1][2]=6;
arr[2][0]=7;
arr[2][1]=8;
arr[2][2]=9;
```

Example

```
//Java Program to illustrate the use of multidimensional array
class Testarray3{
public static void main(String args[]){
//declaring and initializing 2D array
int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
//printing 2D array
for(int i=0;i<3;i++){
 for(int j=0;j<3;j++){
   System.out.print(arr[i][j]+" ");
 }
 System.out.println();
}  }}
```

Output:

```
1 2 3
2 4 5
4 4 5
```

Example

Refer classwork Note

Single Dimensional

1. Sorting an element
2. Searching an element
3. Find maximum and minimum element

Multi-Dimensional

1. Addition of Matrix
2. Subtraction of Matrix
3. Multiplication of Matrix

Matrix Addition

```
import java.util.Scanner;

public class MatrixAddition {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of the
matrices: ");
        int rows = sc.nextInt();
        int cols = sc.nextInt();

        int[][] matrix1 = new int[rows][cols];
        int[][] matrix2 = new int[rows][cols];
        int[][] sum = new int[rows][cols];

        System.out.println("Enter elements of the first matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix1[i][j] = sc.nextInt();
            }
        }

        System.out.println("Enter elements of the second matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix2[i][j] = sc.nextInt();
            }
        }
```

```
                     // Matrix Addition
                     for (int i = 0; i < rows; i++) {
                         for (int j = 0; j < cols; j++) {
                             sum[i][j] = matrix1[i][j] + matrix2[i][j];
                         }
                     }

                     System.out.println("Sum of the matrices:");
                     for (int i = 0; i < rows; i++) {
                         for (int j = 0; j < cols; j++) {
                             System.out.print(sum[i][j] + " ");
                         }
                         System.out.println();
                     }
                 }
}
```

Matrix Subtraction

```
import java.util.Scanner;

public class MatrixAddition {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of the
matrices: ");
        int rows = sc.nextInt();
        int cols = sc.nextInt();

        int[][] matrix1 = new int[rows][cols];
        int[][] matrix2 = new int[rows][cols];
        int[][] sum = new int[rows][cols];

        System.out.println("Enter elements of the first matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix1[i][j] = sc.nextInt();
            }
```

```
          }

          System.out.println("Enter elements of the second matrix:");
          for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
              matrix2[i][j] = sc.nextInt();
            }
          }

          // Matrix Subtraction
          for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
              sum[i][j] = matrix1[i][j] - matrix2[i][j];
            }
          }

          System.out.println("Subtraction of the matrices:");
          for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
              System.out.print(sum[i][j] + " ");
            }
            System.out.println();
          }
        }
}
```

Matrix Multiplication

```
      import java.util.Scanner;

      public class MatrixMultiplication {
        public static void main(String[] args) {
          Scanner sc = new Scanner(System.in);
          System.out.println("Enter the number of rows of the first matrix:
      ");
          int rows1 = sc.nextInt();
          System.out.println("Enter the number of columns of the first
      matrix (also rows of the second matrix): ");
          int cols1 = sc.nextInt();
          System.out.println("Enter the number of columns of the second
      matrix: ");
          int cols2 = sc.nextInt();

          int[][] matrix1 = new int[rows1][cols1];
          int[][] matrix2 = new int[cols1][cols2];
          int[][] product = new int[rows1][cols2];

          System.out.println("Enter elements of the first matrix:");
          for (int i = 0; i < rows1; i++) {
            for (int j = 0; j < cols1; j++) {
              matrix1[i][j] = sc.nextInt();
            }
          }

          System.out.println("Enter elements of the second matrix:");
          for (int i = 0; i < cols1; i++) {
            for (int j = 0; j < cols2; j++) {
              matrix2[i][j] = sc.nextInt();
            }
          }

          // Matrix Multiplication
          for (int i = 0; i < rows1; i++) {
            for (int j = 0; j < cols2; j++) {
              product[i][j] = 0;
              for (int k = 0; k < cols1; k++) {
                product[i][j] += matrix1[i][k] * matrix2[k][j];
              }
            }
          }

          System.out.println("Product of the matrices:");
          for (int i = 0; i < rows1; i++) {
            for (int j = 0; j < cols2; j++) {
              System.out.print(product[i][j] + " ");
            }
            System.out.println();
          }   }}
```