**Name : Thejus Gowda PN**
**kodnestid : KODTMP65S**

# Array Class

The Arrays class in java.util package is a part of the Java Collection Framework. This class provides static methods to dynamically create and access Java arrays. It consists of only static methods and the methods of Object class. The methods of this class can be used by the class name itself.
why do we need java Arrays class when we are able to declare, initialize and compute operations over arrays. The answer to this though lies within the methods of this class which we are going to discuss further as practically these functions help programmers expanding horizons with arrays for instance there are often times when loops are used to do some tasks on an array like:

•Fill an array with a particular value.

•Sort an Arrays.

•Search in an Arrays.

•And many more.

## Methods in Array class

**binarySearch() :**
Searches for the specified element in the array with the help of the Binary Search Algorithm.

```
// Java Program to Demonstrate Arrays Class
// Via binarySearch() method
// Importing Arrays utility class
// from java.util package
import java.util.Arrays;

// Main class
public class GFG {

        // Main driver method
        public static void main(String[] args)
        {
```

```
		// Get the Array
		int intArr[] = { 10, 20, 15, 22, 35 };
		Arrays.sort(intArr);
		int intKey = 22;

		// Print the key and corresponding index
		System.out.println(
			intKey + " found at index = "
			+ Arrays.binarySearch(intArr, intKey));
	}
}
```

output : 22 found at index = 3

**compare(arr1 , arr2) :**
Compares two arrays passed as parameters lexicographically.

If both arrays lexicographically equal returns 0.

I.f the first array is lexicographically smaller than the Second returns -1.

If the Second array is lexicographically smaller than the First
Returns 1.

```
// Java program to demonstrate

// Arrays.compare() method

import java.util.Arrays;

public class Main {

	public static void main(String[] args)

	{


		// Get the Array

		int intArr[] = { 10, 20, 15, 22, 35 };


		// Get the second Array

		int intArr1[] = { 10, 15, 22 };
```

```
                // To compare both arrays
                System.out.println("Integer Arrays on comparison: "
                                                + Arrays.compare(intArr, intArr1));
        }
}
```

```
output : Integer Arrays on comparison: 1
```

**copyOf(originalArray,newLength) :**

Copies the specified array, truncating or padding with the default value (if necessary) so the copy has the specified length.

```
import java.util.Arrays;
public class StudyTonight
{
        public static void main(String[] args)
        {
                int[] arr = new int[] {5, 6, 7, 10, 17};

                System.out.println("Original Array");
                for(int num:arr)
                {
                        System.out.print(num+" ");
                }

                int[] new_arr = Arrays.copyOf(arr, 10);

                System.out.println("\nNew Array");
                for(int num:new_arr)
                {
                        System.out.print(num+" ");
                }
        }
}
```

output :

```
Original Array :  5 6 7 10 17
New Array :  5 6 7 10 17 0 0 0 0 0
```

**toString(originalArray) :**
It returns a string representation of the contents of this array. The string representation consists of a list of the array’s elements, enclosed in square brackets (“[]”). Adjacent elements are separated by the characters a comma followed by a space. Elements are converted to strings as by String.valueOf() function.

```
// Java program to demonstrate
// Arrays.toString() method

import java.util.Arrays;

public class Main {
	public static void main(String[] args)
	{

		// Get the Array
		int intArr[] = { 10, 20, 15, 22, 35 };

		// To print the elements in one line
		System.out.println("Integer Array: "
						+ Arrays.toString(intArr));
	}
}
```

output : Integer Array: [10, 20, 15, 22, 35]

**sort(originalArray) :**

Sorts the complete array in ascending order.

```
// Java program to demonstrate
// Arrays.sort() method

import java.util.Arrays;

public class Main {
	public static void main(String[] args)
	{

		// Get the Array
		int intArr[] = { 10, 20, 15, 22, 35 };

		// To sort the array using normal sort-
		Arrays.sort(intArr);
```

```
                System.out.println("Integer Array: "
                                + Arrays.toString(intArr));
        }
}
```

output : Integer Array: [10, 15, 20, 22, 35]

**equals(array1 , array2) :**

Checks if both the arrays are equal or not.

```
// Java program to demonstrate
// Arrays.equals() method

import java.util.Arrays;

public class Main {
        public static void main(String[] args)
        {

                // Get the Arrays
                int intArr[] = { 10, 20, 15, 22, 35 };

                // Get the second Arrays
                int intArr1[] = { 10, 15, 22 };

                // To compare both arrays
                System.out.println("Integer Arrays on comparison: "
                                + Arrays.equals(intArr, intArr1));
        }
}
```

output : Integer Arrays on comparison: false

**hashCode(originalArray) :**
Returns an integer hashCode of this array instance.

```
// Java program to demonstrate
// Arrays.hashCode() method

import java.util.Arrays;

public class Main {
	public static void main(String[] args)
	{

		// Get the Array
		int intArr[] = { 10, 20, 15, 22, 35 };

		// To get the hashCode of the arrays
		System.out.println("Integer Array: "
						+ Arrays.hashCode(intArr));
	}
}
```

```
Integer Array: 38475313
```