Introduction

In this article, we are going to learn arrays class in Java. Arrays class is present in the java.util package. Arrays class concept comes under the collection framework in java. Arrays class provides several utility methods. Using the Arrays class, we can perform searches and can sort and accomplish some other tasks in easy ways.

Prerequisites:

* Basic knowledge of java
* Basic knowledge of array in java. You can learn ***How to use Array in Java*** by [clicking here](https://www.c-sharpcorner.com/article/java-array-tutorial/).

Arrays Class Methods

There are many methods that are used with arrays class. Here we are going to learn some of the important methods.

1. as List() Method

This method will return a list. We have to pass an array object as a parameter in this method.

**Example**

Import java.util.\*;

Class test {

Public static void main(String args[]) {

//Int List

Integer [] arr1 = new Integer [] {

10,

20,

30,

40,

50

};

List < Integer > list1 = Arrays.asList (arr1);

System.out.println ("Integer List:" + list1);

//Float list

Float [] arr2 = new Float [] {

10.0 f, 20.0 f, 30.0 f, 40.0 f, 50.0 f

};

List < Float > list2 = Arrays.asList (arr2);

System.out.println ("Float List:" + list2);

//String list

String arr3[] = new String[] {

"C",

"C++",

"Java",

"Python"

};

List < String > list3 = Arrays.asList (arr3);

System.out.println ("String List" + list3);

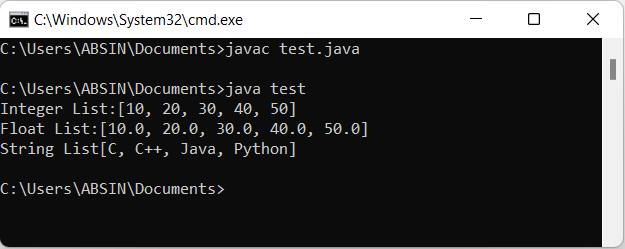
}

}

Java

Copy

**Output**



**Important Point**

When we use the as List () method, we have to use Integer instead of int. If we use int, then we will get an error. The same rule applies to others also.

2. Sort () Method

This method is used to sort an array in ascending or descending order.

**Example**

Import java.util.\*;

Class test {

Public static void main (String args []) {

Integer arr1 [] = new Integer [] {

50,

40,

30,

20,

10

};

// Sort Integer Array

Arrays.sort (arr1);

For (int i = 0; i < arr1.length; i++) {

System.out.print (arr1[i] + " ");

}

//For the next line

System.out.println ();

Float arr2 [] = new Float [] {

50.0 f, 40.0 f, 30.0 f, 20.0 f, 10.0 f

};

// Sort Float Array

Arrays. Sort (arr2);

For (int i = 0; i < arr2.length; i++) {

System.out.print (arr2 [i] + " ");

}

//For the next line

System.out.println ();

String arr3[] = new String[] {

"Python",

"Java",

"C",

"C++"

};

//Sort String Array

Arrays. Sort (arr3);

For (int i = 0; i < arr3.length; i++) {

System.out.print (arr3[i] + " ");

}

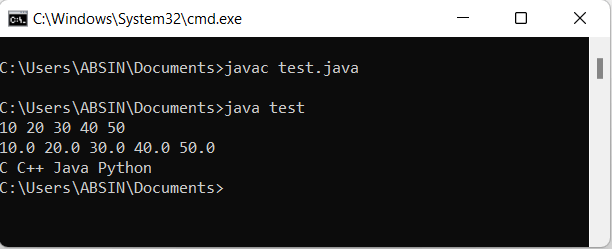
}

}

Java

Copy

**Output**

****

3. Binary Search () Method

This method is used to search for an element inside a sorted array. We can apply a binary search on an array, but the array must be in sorted order. In this method, we need to pass two parameters. The first parameter is a sorted array object, and the second is a key. A key is the element that we want to search inside an array. This method will return the index value.

**Example**

Import java.util.\*;

Class test {

Public static void main (String args []) {

Integer arr1[] = new Integer[] {

10,

20,

30,

40,

50

};

Int key = 30;

Int index = Arrays.binarySearch (arr1, key);

System.out.print (key + " found at index: " + index);

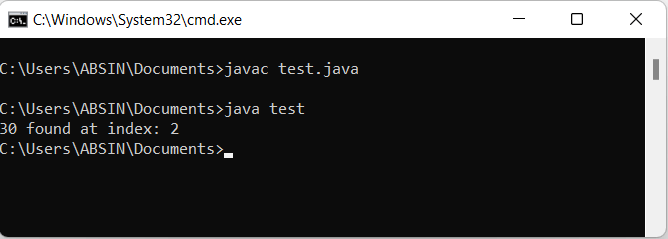
}

}

Java

Copy

**Output**



4. Equals () Method

This method is used to compare two arrays and check whether both arrays are equal or not.

**Example**

Import java.util.\*;

Class test {

Public static void main(String args[]) {

Integer arr1[] = new Integer[] {

10,

20,

30,

40,

50

};

Integer arr2[] = new Integer[] {

10,

20,

30,

40,

60

};

if (Arrays.equals(arr1, arr2)) {

System.out.println("Both Arrays Are Equal");

} else {

System.out.println("Both Arrays Are Not Equal");

}

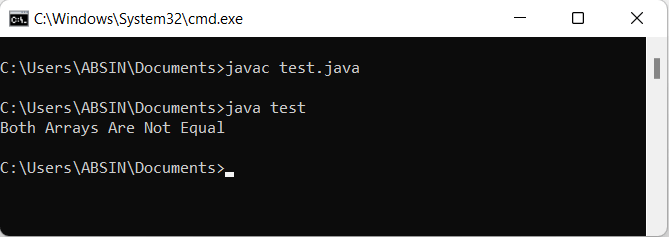
}

}

Java

Copy

**Output**

****

5. compare() Method

This method is used to compare two arrays. If both arrays are equal then it will return a 0 value. Otherwise, it will return a 1 or -1 value.

**Example**

import java.util.\*;

class test {

public static void main(String args[]) {

Integer arr1[] = new Integer[] {

10,

20,

30,

40,

50

};

Integer arr2[] = new Integer[] {

10,

20,

30,

40,

60

};

if (Arrays.compare(arr1, arr2) == 0) {

System.out.println("Both arrays are equal");

} else {

System.out.println("Both arrays are not equal");

}

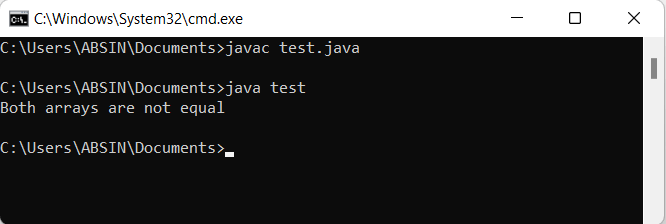
}

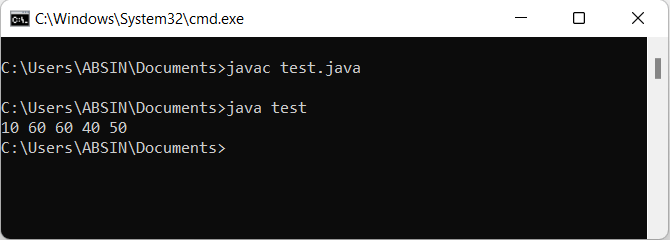
}

Java

Copy

**Output**

****

****