JAVA.UTIL.ARRAYS.COPYOFRANGE*LONG[], INT, INT* METHOD

Description

The **java.util.Arrays.copyOfRange*long[]original, intfrom, intto*** method copies the specified range of the specified array into a new array.The final index of the range *to*, which must be greater than or equal to from, may be greater than original.length, in which case 0L is placed in all elements of the copy whose index is greater than or equal to original.length - from. The length of the returned array will be to - from.

Declaration

Following is the declaration for **java.util.Arrays.copyOfRange** method

```
public static long[] copyOfRange(long[] original, int from, int to)
```

Parameters

- **original** -- This is the array from which a range is to to be copied.
- **from** -- This is the initial index of the range to be copied, inclusive.
- **to** -- This is the final index of the range to be copied, exclusive.

Return Value

This method returns a new array containing the specified range from the original array, truncated or padded with zeros to obtain the required length.

Exception

- **ArrayIndexOutOfBoundsException** -- If from < 0 or from > original.length
- **IllegalArgumentException** -- If from > to.
- **NullPointerException** -- If original is null.

Example

The following example shows the usage of java.util.Arrays.copyOfRange method.

```
package com.tutorialspoint;

import java.util.Arrays;

public class ArrayDemo {

   public static void main(String[] args) {

   // intializing an array arr1
   long[] arr1 = new long[] {5, 62, 15};

   // printing the array
   System.out.println("Printing 1st array:");
   for (int i = 0; i < arr1.length; i++) {
   System.out.println(arr1[i]);
   }

   // copying array arr1 to arr2 with range of index from 2 to 5
   long[] arr2 = Arrays.copyOfRange(arr1, 2, 5);

   // printing the array arr2
   System.out.println("Printing new array:");
```

```
    for (int i = 0; i < arr2.length; i++) {
    System.out.println(arr2[i]);
    }
    }
}
```

Let us compile and run the above program, this will produce the following result:

```
Printing 1st array:
5
62
15
Printing new array:
15
0
0
```