**آية خالد أبو الرجال**

**علوم حاسوب**

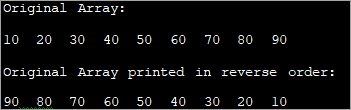
1. Write a Java method that Reverse an array using another array.

**Let’s consider an array as shown below:**

**The following program prints the array in reverse order.**

|  |
| --- |
| import java.util.\*;  import java.util.stream.\*;  public class Main  {      public static void main(String[] args) {      Integer[] intArray = {10,20,30,40,50,60,70,80,90};      //print array starting from first element      System.out.println("Original Array:");      for(int i=0;i<intArray.length;i++)           System.out.print(intArray[i] + "  ");        System.out.println();        //print array starting from last element      System.out.println("Original Array printed in reverse order:");           for(int i=intArray.length-1;i>=0;i--)           System.out.print(intArray[i] + "  ");      }  } |

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2020/01/Printing-array-in-Reverse-Order.png)

1. Write a Java method that Clone an array to a backup array.

|  |
| --- |
| // A Java program to demonstrate array copy using clone()  public class Test {      public static void main(String[] args)      {          int a[] = { 1, 8, 3 };            // Copy elements of a[] to b[]          int b[] = a.clone();            // Change b[] to verify that          // b[] is different from a[]          b[0]++;          System.out.println("Contents of a[] ");          for (int i = 0; i < a.length; i++)              System.out.print(a[i] + " ");          System.out.println("\n\nContents of b[] ");          for (int i = 0; i < b.length; i++)              System.out.print(b[i] + " ");      }  } |

**Output**

Contents of a[]

1 8 3

Contents of b[]

2 8 3

1. Write a Java method that remove elements from an array.

### Using Two Arrays

The simplest pure Java way to do this is to make a new array, one element shorter than the original one and copy all element, except the one we'd like to remove, into it:

int[] copy = new int[array.length - 1];

for (int i = 0, j = 0; i < array.length; i++) {

if (i != index) {

copy[j++] = array[i];

}

}

Here, we're simply iterating over the original array and copying elements from the original array into the new array, skipping the one we'd like to remove.

The copy array now consists of:

10, 20, 30, 50, 60, 70, 80, 90, 100

### ArrayUtils.remove()

In case you're already using the Apache Commons library, you can use the ArrayUtils.remove() method.

Before working with Apache Commons, we'll want to add it to our project:

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

<version>${version}</version>

</dependency>

Using the method is really simple. We simply supply it with the array we'd like to remove an element from and its index:

int[] array = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};

int index = 3;

array = ArrayUtils.remove(array, index);

It then returns the new array, which is stored in the array variable:

10, 20, 30, 50, 60, 70, 80, 90, 100

1. Write a Java method that repeatedly selects and removes a random entry from an array until the array holds no more entries.

import java.util.Random;  
  
/\*\*  
\* Created by sakura on 9/19/16.  
\*/  
public class Main {  
public static void main(String[] args) {  
int[] array = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };  
removeElements(array);  
}  
  
static void removeElements(int[] array) {  
Random r = new Random();  
while (array.length > 0) {  
int index = r.nextInt(array.length);  
System.out.println("INDEX = " + index + ", ELEMENT = " + array[index]);  
int[] array1 = new int[array.length - 1];  
for (int i = 0; i < index; i++)  
array1[i] = array[i];  
for (int i = index; i < array.length - 1; i++)  
array1[i] = array[i + 1];  
array = array1;  
}  
}  
}