**Can we use = to copy an array into another array?**

We can’t use = to copy an array to another array. Arrays are mutable and when we use = to copy an array, it won’t really copy the array but the second array points to the same memory area of the first array. Since arrays are mutable, if we change a value in an array it changes the value in second array also.

Example:

**package** package1;

**import** java.util.Arrays;

**public** **class** ArraysExample {

**public** **static** **void** main(String[] args) {

String arr1[] = {"subbu", "selenium", "java", "tutorials"};

String arr2[] = **new** String[5];

arr2 = arr1;

arr2[2] = "hello";

System.***out***.println(Arrays.*toString*(arr1));

System.***out***.println(Arrays.*toString*(arr2));

}

}

Result:

[subbu, selenium, hello, tutorials]

[subbu, selenium, hello, tutorials]

**How to copy an array to another array?**

1. **Object.clone()**
2. **System.arraycopy()**
3. **Arrays.copyOf()**
4. **Arrays.copyOfRange()**

**You can copy an array using Object.clone() method.**

**package** package1;

**import** java.util.Arrays;

**public** **class** ArraysExample {

**public** **static** **void** main(String[] args) {

String arr1[] = {"subbu", "selenium", "java", "tutorials"};

String arr2[] = **new** String[5];

arr2 = arr1.clone();

arr2[2] = "hello";

System.***out***.println(Arrays.*toString*(arr1));

System.***out***.println(Arrays.*toString*(arr2));

}

}

**You can copy an array using System.arraycopy() method.**

System.arraycopy() takes the following arguments.

First argument: Source array

Second argument: Index from where copying should start in the source array

Third argument: Destination array

Fourth argument: Index from where the elements should be copied to, in the destination array

Fifth argument: No. of elements to be copied.

**package** package1;

**import** java.util.Arrays;

**public** **class** ArraysExample {

**public** **static** **void** main(String[] args) {

String arr1[] = {"subbu", "selenium", "java", "tutorials"};

String arr2[] = **new** String[4];

System.*arraycopy*(arr1, 0, arr2, 0, 4);

System.***out***.println(Arrays.*toString*(arr1));

System.***out***.println(Arrays.*toString*(arr2));

}

}

**You can copy an array using Arrays.copyOf() method.**

Arrays.copyOf() method takes two arguments.

First Argument: This is the source array

Second Argument: No. of elements that should be copied from source array starting from 0 index

**package** package1;

**import** java.util.Arrays;

**public** **class** ArraysExample {

**public** **static** **void** main(String[] args) {

String arr1[] = {"subbu", "selenium", "java", "tutorials"};

String arr2[] = **new** String[4];

arr2 = Arrays.*copyOf*(arr1, 3);

System.***out***.println(Arrays.*toString*(arr1));

System.***out***.println(Arrays.*toString*(arr2));

}

}

**You can copy an array using Arrays.copyOfRange() method.**

Arrays.copyOfRange() method takes three arguments.

First Argument: Source Array

Second Argument: The starting index from where elements should be copied

Third Argument: The ending index of the elements that should be copied. This is not included when the elements are copied.

**package** package1;

**import** java.util.Arrays;

**public** **class** ArraysExample {

**public** **static** **void** main(String[] args) {

String arr1[] = {"subbu", "selenium", "java", "tutorials"};

String arr2[] = **new** String[4];

arr2 = Arrays.*copyOfRange*(arr1, 1, 3);

System.***out***.println(Arrays.*toString*(arr1));

System.***out***.println(Arrays.*toString*(arr2));

}

}