**Array in java**

Point-1: int arr[];array is not created only reference created

arr[0]=10;In java, Array is created using 'new' created

arr[1]=40;

Point-2: int arr\_1[]=new int[5];

int arr\_2[]=new int[]{11,22,33,44,55};

**int arr\_3[]={12,23,22,33,44};**

int arr\_4[]=new int[5]{2,4,5,2,3};//Error

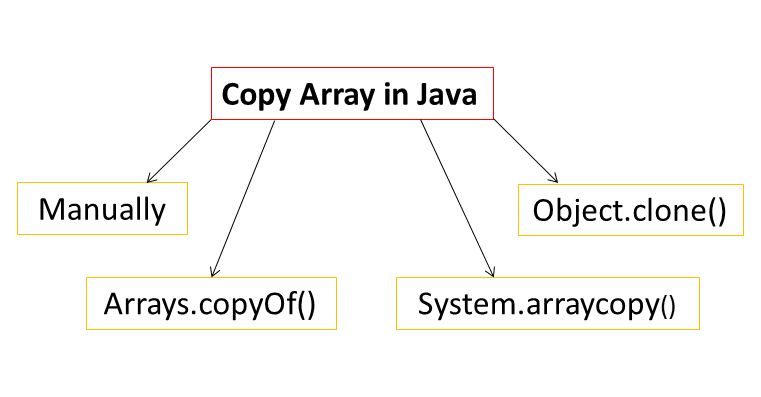
// you can not mention size and value together

**1. int** a[]=**new** **int**[5];//declaration and instantiation

a[0]=10;//initialization

**2.int** a[]={33,3,4,5};//declaration, instantiation and initialization

**Different way to Copy one Array into Other:**



int a[] = {1, 8, 3};

int b[] = new int[a.length];

b = a;

we actually assigning reference of array. Hence if we make any change to one array, it would be reflected in other array

**1. Manually**

int a[]={10,20,30,40,50};

int b[]=new int[a.length];

for(int i=0;i<a.length;++i){

**b[i]=a[i];**

}

**2. Clone**

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.arraycopy()**

int a[] = {1, 8, 3};

// Create an array b[] of same size as a[]

int b[] = new int[a.length];

// Copy elements of a[] to b[]

**System.arraycopy(a, 0, b, 0, 3);**

**Arrays.copyOf()**

int a[]={10,20,30,40,50};

int b[]=new int[a.length];

//copying one array to another

**b=Arrays.copyOf(a,a.length);**

class Test

{

    public static void main (String[] args)

    {

        int arr1[] = {1, 2, 3};

        int arr2[] = {1, 2, 3};

        if (arr1 == arr2)

            System.out.println("Same");

        else

            System.out.println("Not same");

    }

}

Output:

Not Same

|  |
| --- |
| import java.util.Arrays;  class Test  {      public static void main (String[] args)      {          int arr1[] = {1, 2, 3};          int arr2[] = {1, 2, 3};          if (Arrays.equals(arr1, arr2))              System.out.println("Same");          else              System.out.println("Not same");      }  } |

Output:

Same

|  |
| --- |
| import java.util.Arrays;  class Test  {      public static void main (String[] args)      {          // inarr1 and inarr2 have same values          int inarr1[] = {1, 2, 3};          int inarr2[] = {1, 2, 3};          Object[] arr1 = {inarr1};  // arr1 contains only one element          Object[] arr2 = {inarr2};  // arr2 also contains only one element          if (Arrays.equals(arr1, arr2))              System.out.println("Same");          else              System.out.println("Not same");      }  } |

Output:

Not Same

|  |
| --- |
| import java.util.Arrays;  class Test  {      public static void main (String[] args)      {          int inarr1[] = {1, 2, 3};          int inarr2[] = {1, 2, 3};          Object[] arr1 = {inarr1};  // arr1 contains only one element          Object[] arr2 = {inarr2};  // arr2 also contains only one element          if (Arrays.deepEquals(arr1, arr2))              System.out.println("Same");          else              System.out.println("Not same");      }  } |

Output:

Same

Copying a Java Array

We can copy an array to another by the arraycopy() method of System class.

**Syntax of arraycopy method:**

**public** **static** **void** arraycopy( Object src, **int** srcPos,Object dest, **int** destPos, **int** length)

**class** TestArrayCopyDemo {

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

        //declaring a source array

**char**[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e', 'i', 'n', 'a', 't', 'e', 'd' };

        //declaring a destination array

**char**[] copyTo = **new** **char**[7];

        //copying array using System.arraycopy() method

**System.arraycopy(**copyFrom, 2, copyTo, 0, 7);

        //printing the destination array

        System.out.println(String.valueOf(copyTo));

OR

// Print elements of destination  using loop

    }

}

Output:

caffein