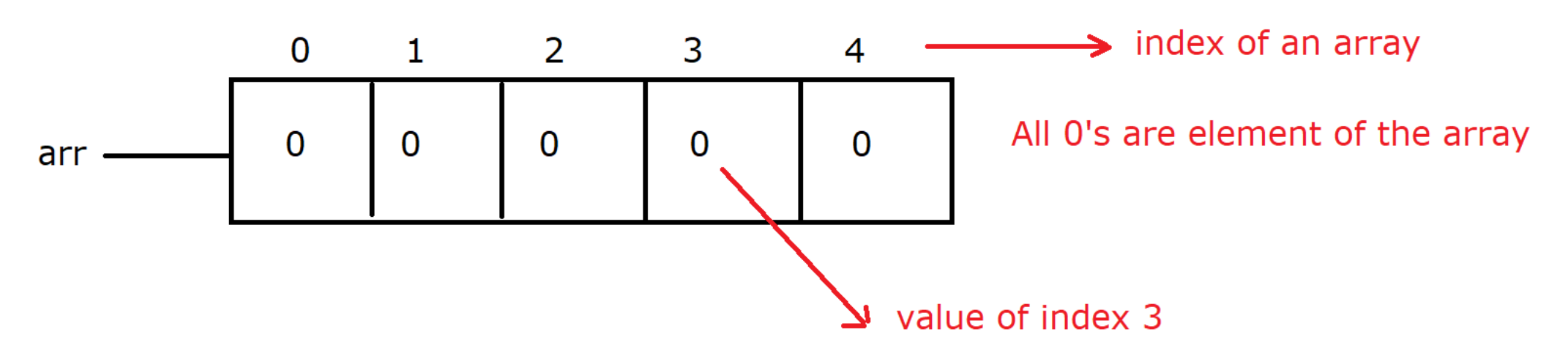
What is an array in java ?
----
* An array is a **container object** which is used to hold **fixed** number of values in an **ordered** Collection.
* An array is an object in java so, It is always created inside **HEAP MEMORY.**
* The length of an array is established when the array is created. After creation, its length is fixed.

Example :
----
int []arr = new int[5];



Characteristics of an Array :
----
1) Homogeneous Elements:

All elements in an array must be of the same data type but by using upcasting we can accept Hetrogeneous elements also.

2) Fixed Size :

Once an array is created, its size cannot be changed.(size is nothing but number of elements are available in the array)

3) Indexed Based:

We can store and access the elements using index (arr[0] for the first element, arr[1] for the second, etc.)

a) How to find first and last index position element
size = 10; [length = 10]

int []arr = {10,20,30,40,90,67,34};

First Index : arr[0]
Last Index : arr[arr.length -1]

b) [Why array index starts from 0] [int is taking 4 bytes of Memory]
1000 + 0 * 4 = 1000x
1000 + 1 * 4 = 1004X
1000 + 2 * 4 = 1008x

4) Stored in Contiguous Memory:

Array elements are stored in adjacent memory locations, improving access speed (CACHE Memory).

5) Stored in Heap Memory:

Arrays are objects in Java and are stored in the HEAP MEMORY.

6) Default Values:

If an array is declared but not initialized, Java assigns default values because it is a reference data type.

7) Efficient Retrieval:

Direct access to elements using an index provides fast data retrieval (O(1) time complexity).

Drawback of an array :
----
1) Fixed in size
2) Insertion OR deletion at the begning or middle is O(n)
3) No predefined method support.

Types of Array in Java :
----
* In Java, Basically We have 2 types of array :

1) Single OR One Dimensional Array.([])
2) 2D OR Multi-Dimensional Array.([][])

Note :We have jagged array which is a multi-dimensional array where each row can have a different number of columns

----
Working with single OR one Dimensional Array :

In Java we can create single dimensional array by using following ways :

1) The most common way to create an array object by using new keyword :

Example :

int [] arr = new int[5];

An array is an Object in java so whenever we create an array in java then Internally JVM provides a class which is loaded into JVM memory on the behalf of Array.

int x; //Primitive
int []x; //Object

Primitive type Array	JVM Provided class (JNI ENCODING)
int []	[I
byte[]	[B
short[]	[S
long[]	[J
flaot[]	[F
double[]	[D
char[]	[C
boolean[]	[Z
String[]	[LFully Qualified Name

2) Inline Initialization :

Another way to create an array object using new keyword with Inline Initialization
[We can initialize the array directly while creating it.]

Example :
int[] arr = new int[]{10, 20, 30, 40, 50};

Note : Here instead of default value user provides the value explicitly.

In the above example, expression is duplicate because the type of array we know from the left side of equal sign and since we are specifying the value of array hence we know the size also.

Instead of this we can use more convenient way which is known as Anonymous array.

3) Anonymous Array :

Example 3 :
----
int[] arr = {100, 200, 300};

Note : This is called anonymous array becuase we don't specify type and size.

Finally, We can use [] before or after the name, providing space is optional. The following 5 statements are same

int[] num1;
int [] num2;
int []num3;
int num4[];
int num5 [];

Q1) WAP to show JNI coding standard.

```
package com.ravi.basic;

public class ArrayEx1 {

    public static void main(String[] args)
    {
        byte[] arr1 = new byte[5];
        System.out.println(arr1.getClass().getName());

        short[] arr2 = new short[5];
        System.out.println(arr2.getClass().getName());


        int[] arr3 = new int[5];
        System.out.println(arr3.getClass().getName());

        long[] arr4 = new long[5];
        System.out.println(arr4.getClass().getName());


        float[] arr5 = new float[5];
        System.out.println(arr5.getClass().getName());

        double[] arr6 = new double[5];
        System.out.println(arr6.getClass().getName());

        char[] arr7 = new char[5];
        System.out.println(arr7.getClass().getName());

        boolean[] arr8 = new boolean[5];
        System.out.println(arr8.getClass().getName());


        Integer[] arr9 = new Integer[5];
        System.out.println(arr9.getClass().getName());

    }

}
```

Note : From the above program, It is clear that Internally JNI classes are working which is provided by JVM (Loaded by JVM when we create the Object)
All these classes are exteding from Object class and implements Cloneable and Serializable

int []arr = new int[5];

class I extends Object implements Cloneable, Serializabe
{
}

Q2) Different ways to create 1D array and fetching the elements using ordinary for loop /forEach loop and toString().

public static String toString(int[] a) :
----
* This method is available from JDK 1.5 to convert int[] array into String.

```
package com.ravi.basic;

import java.util.Arrays;

public class ArrayEx2
{
    public static void main(String[] args)
    {
        int []arr = new int[3];
        arr[0] = 100;
        arr[1] = 200;
        arr[2] = 300;

        System.out.println("Fetching the Array Data using Ordinary for Loop :");
        for(int i=0; i<arr.length; i++)
        {
            System.out.println(arr[i]);
        }


        int []x = new int[] {1000,2000,3000};
        System.out.println("Fetching the Array Data using for each Loop :");

        for(int data : x)
        {
            System.out.println(data);
        }

        int []arr3 = {2,9,4,8, 6};
        System.out.println(Arrays.toString(arr3));

    }
}
```