



Starts @ 9:05 pm

---

| What is an array in Java?

primitive data type  
int, float, double, boolean - large

String →

same type  
array = 5

An array is a finite and ordered collection of homogeneous data elements. It is a data structure that contains a limited number of elements in ordered because all the elements are stored one by one in contiguous locations of computer memory. In a linear fashion, it is homogeneous because all elements of an array are of the same data type only. We can store either primitive types or object references into it.

int arr(); → heap

arr = (int) int[5];

Why indexing starts from zero?  
0 base for you are from your base address 9

$$y = 1$$

$$y = 2$$

$$y = 3$$

$$y = 4$$

$$y = 5$$

$$y = 6$$

$$y = 7$$

$$y = 8$$

$$y = 9$$

$$y = 10$$

$$y = 11$$

$$y = 12$$

$$y = 13$$

$$y = 14$$

$$y = 15$$

$$y = 16$$

$$y = 17$$

$$y = 18$$

$$y = 19$$

$$y = 20$$

$$y = 21$$

$$y = 22$$

$$y = 23$$

$$y = 24$$

$$y = 25$$

$$y = 26$$

$$y = 27$$

$$y = 28$$

$$y = 29$$

$$y = 30$$

$$y = 31$$

$$y = 32$$

$$y = 33$$

$$y = 34$$

$$y = 35$$

$$y = 36$$

$$y = 37$$

$$y = 38$$

$$y = 39$$

$$y = 40$$

$$y = 41$$

$$y = 42$$

$$y = 43$$

$$y = 44$$

$$y = 45$$

$$y = 46$$

$$y = 47$$

$$y = 48$$

$$y = 49$$

$$y = 50$$

$$y = 51$$

$$y = 52$$

$$y = 53$$

$$y = 54$$

$$y = 55$$

$$y = 56$$

$$y = 57$$

$$y = 58$$

$$y = 59$$

$$y = 60$$

$$y = 61$$

$$y = 62$$

$$y = 63$$

$$y = 64$$

$$y = 65$$

$$y = 66$$

$$y = 67$$

$$y = 68$$

$$y = 69$$

$$y = 70$$

$$y = 71$$

$$y = 72$$

$$y = 73$$

$$y = 74$$

$$y = 75$$

$$y = 76$$

$$y = 77$$

$$y = 78$$

$$y = 79$$

$$y = 80$$

$$y = 81$$

$$y = 82$$

$$y = 83$$

$$y = 84$$

$$y = 85$$

$$y = 86$$

$$y = 87$$

$$y = 88$$

$$y = 89$$

$$y = 90$$

$$y = 91$$

$$y = 92$$

$$y = 93$$

$$y = 94$$

$$y = 95$$

$$y = 96$$

$$y = 97$$

$$y = 98$$

$$y = 99$$

$$y = 100$$

$$y = 101$$

$$y = 102$$

$$y = 103$$

$$y = 104$$

$$y = 105$$

$$y = 106$$

$$y = 107$$

$$y = 108$$

$$y = 109$$

$$y = 110$$

$$y = 111$$

$$y = 112$$

$$y = 113$$

$$y = 114$$

$$y = 115$$

$$y = 116$$

$$y = 117$$

$$y = 118$$

$$y = 119$$

$$y = 120$$

$$y = 121$$

$$y = 122$$

$$y = 123$$

$$y = 124$$

$$y = 125$$

$$y = 126$$

$$y = 127$$

$$y = 128$$

$$y = 129$$

$$y = 130$$

$$y = 131$$

$$y = 132$$

$$y = 133$$

$$y = 134$$

$$y = 135$$

$$y = 136$$

$$y = 137$$

$$y = 138$$

$$y = 139$$

$$y = 140$$

$$y = 141$$

$$y = 142$$

$$y = 143$$

$$y = 144$$

$$y = 145$$

$$y = 146$$

$$y = 147$$

$$y = 148$$

$$y = 149$$

$$y = 150$$

$$y = 151$$

$$y = 152$$

$$y = 153$$

$$y = 154$$

$$y = 155$$

$$y = 156$$

$$y = 157$$

$$y = 158$$

$$y = 159$$

$$y = 160$$

$$y = 161$$

$$y = 162$$

$$y = 163$$

$$y = 164$$

$$y = 165$$

$$y = 166$$

$$y = 167$$

$$y = 168$$

$$y = 169$$

$$y = 170$$

$$y = 171$$

$$y = 172$$

$$y = 173$$

$$y = 174$$

$$y = 175$$

$$y = 176$$

$$y = 177$$

$$y = 178$$

$$y = 179$$

$$y = 180$$

$$y = 181$$

$$y = 182$$

$$y = 183$$

$$y = 184$$

$$y = 185$$

$$y = 186$$

$$y = 187$$

$$y = 188$$

$$y = 189$$

$$y = 190$$

$$y = 191$$

$$y = 192$$

$$y = 193$$

$$y = 194$$

$$y = 195$$

$$y = 196$$

$$y = 197$$

$$y = 198$$

$$y = 199$$

$$y = 200$$

$$y = 201$$

$$y = 202$$

$$y = 203$$

$$y = 204$$

$$y = 205$$

$$y = 206$$

$$y = 207$$

$$y = 208$$

$$y = 209$$

$$y = 210$$

$$y = 211$$

$$y = 212$$

$$y = 213$$

$$y = 214$$

$$y = 215$$

$$y = 216$$

$$y = 217$$

$$y = 218$$

$$y = 219$$

$$y = 220$$

$$y = 221$$

$$y = 222$$

$$y = 223$$

$$y = 224$$

$$y = 225$$

$$y = 226$$

$$y = 227$$

$$y = 228$$

$$y = 229$$

$$y = 230$$

$$y = 231$$

$$y = 232$$

$$y = 233$$

$$y = 234$$

$$y = 235$$

$$y = 236$$

$$y = 237$$

$$y = 238$$

$$y = 239$$

$$y = 240$$

$$y = 241$$

$$y = 242$$

$$y = 243$$

$$y = 244$$

$$y = 245$$

$$y = 246$$

$$y = 247$$

$$y = 248$$

$$y = 249$$

$$y = 250$$

$$y = 251$$

$$y = 252$$

$$y = 253$$

$$y = 254$$

$$y = 255$$

$$y = 256$$

$$y = 257$$

$$y = 258$$

$$y = 259$$

$$y = 260$$

$$y = 261$$

$$y = 262$$

$$y = 263$$

$$y = 264$$

$$y = 265$$

$$y = 266$$

$$y = 267$$

$$y = 268$$

$$y = 269$$

$$y = 27$$

What are the types of an array?

- ↳ Single dimensional (1D)
- ↳ Multidimensional array (2D, 3D, 4D, ...)

Arrays are generally categorized into two parts as described below

- Single Dimensional Array
- Multi-Dimensional Array (2D and 3D arrays)

Is it possible to declare array size as negative?

↗ No

No, it is not possible to declare array size as negative. Still, if we declare the negative size, there will be no compile-time error. But we get the NegativeArraySizeException at run-time.

```
int arr() = new int(-5);
```

## What is the default value of the array?

- ↳ int, byte, short, long → 0
  - ↳ float, double → 0.0
  - ↳ object / string → null
  - ↳ boolean - false
- ↳ primitive data type  
↳ default = initialize
- non-primitive  
~~↳ null~~

When we create a new array, it always initialized with the default values. The default values of the array are: If an array is of byte, short, int, and long type, the default value is 0. If an array is of float and double type, the default value is 0. If an array is of Boolean type, the default value is false. If an array is of an Object type, the default value is null.

On which memory arrays are created in Java?

Stack.

*heap*

Arrays are created on dynamic memory by JVM. There is no question of static memory in Java everything (variable, array, object, etc.) is created on dynamic memory only.

fixed-size

Can we modify the size (or length) of the array once set it?

↳ No

No. Once the size of an array is defined, we cannot modify it. We cannot add or delete elements after creation of an array. In other words, an array cannot expand or shrink at runtime.

What is **ArrayIndexOutOfBoundsException** in java? When it occurs?

index not exist

ArrayIndexOutOfBoundsException is a  
run time exception which occurs when  
your program tries to access invalid  
index of an array i.e negative index or  
index higher than the size of the array.

\* → -ve index.

→ higher than size of array.

# What is Object Oriented Programming?

Object-Oriented Programming(OOPs) is a type of programming that is based on objects rather than just functions and procedures. Individual objects are grouped into classes. OOPs implements real-world entities like inheritance, polymorphism, hiding etc into programming. It also allows binding data and code together.

encapsulation

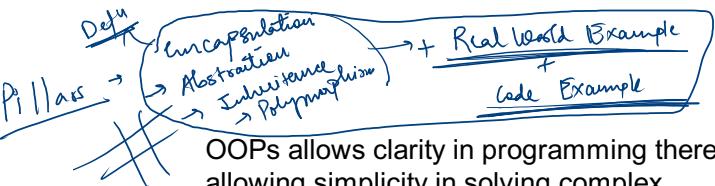
# What is a Constructor?

~~Learn it~~

A constructor in Java is similar to a method that is invoked when an object of the class is created. Unlike Java methods, a constructor has the same name as that of the class and does not have any return type.

## Why we use OOPs?

- ↳ Polymorphism
- Encapsulation
- Abstraction
- Inheritance



OOPs allows clarity in programming thereby allowing simplicity in solving complex problems Code can be reused through inheritance thereby reducing redundancy

Data and code are bound together by encapsulation OOPs allows data hiding, therefore, private data is kept confidential. Problems can be divided into different parts making it simple to solve. The concept of polymorphism gives flexibility to the program by allowing the entities to have multiple forms

# What are the main features of OOPs?

↳ 4 pillars

Inheritance  
Encapsulation  
Polymorphism  
Data Abstraction



---

---

What is an object?

---

~~Learn it~~

- An object is a real-world entity which is the basic unit of OOPs for example chair, cat, dog, etc.
- Different objects have different states or attributes, and behaviors.

What is a class?

A class is a prototype that consists of objects in different states and with different behaviors. It ~~has a number of methods that are common to the objects present within that class.~~

What is the difference between int array[] and int[] array?

↳ ND

There is no difference between array[] and []array.  
Both array[] and []array are the ways to declare an array. The only difference between them is that if we are declaring more than one array in a line, we should use prefix []. If we are declaring a single array in a line, we should use postfix [].

int arr(), arr2(),

~~int arr, arr2;~~

---

## How to copy an array in Java?

---

ao<sup>x</sup>

We can create a copy of an array in two ways,  
first one is manually by iterating over the array  
and the second one is by using the arrayCopy()  
method. Using the arrayCopy() method of the  
System class is the fastest way to copy an array  
and also allows us to copy a part of the array.  
These two methods are the popular ways to  
copy an array. The other two methods to copy an  
array is to use the Arrays.copyOf() method and  
using clone() method.

```

↳ arrayCopy()
↳ clone()
→ Arrays.copyOf()

```

$\text{arr2}[2] = -2$   
 $\text{arr2}(\text{arr1}[1]) \rightarrow -10$   
 $\text{Supr}(\text{arr2})$   
 $\text{Supr}(\text{arr2}[2])$   
 $\hookrightarrow -7$   
 $\text{arr1}[1] = -10$

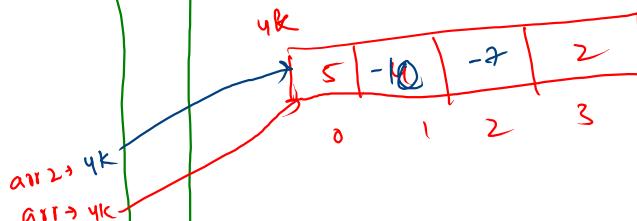
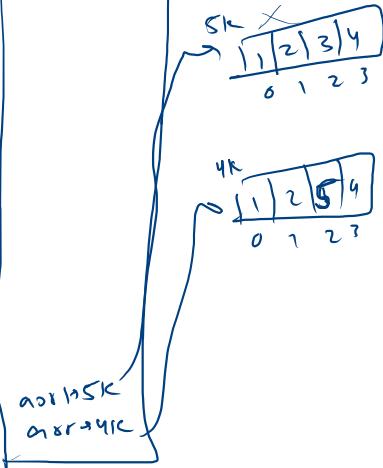
```

int arr1[] = new int [arr.length];
for(int i = 0; i < arr.length; i++)
    arr1[i] = arr[i];

```

$$\text{arr}[2] = 5$$

~~int arr2[] = arr;~~  
 $\downarrow$   
 copying address here





What happens if we declare an array without assigning the size?

```
int arr[];  
arr = new int(size);
```

It is not possible to declare an array without size. When we declare an array without assigning the size, it throws the compile-time error. For example, height=new int[].

---

What is the difference between Array and ArrayList?

---

~~ArrayList < int > X~~  
~~ArrayList < Integer >~~

Array: Array is static. It is of fixed size. Its size cannot be changed once it is declared. It contains both primitive data types and objects of a class. Array does not have generic features.) ArrayList: ArrayList is dynamic in size. Its size or capacity automatically grows when we add element into it. It contains only the object entries. It has a generic feature.

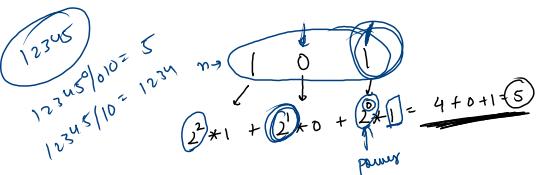
Name the keyword that is used for allocating memory space to an array?

int arr() - new int [size]  
+  
heap  
~~stack~~

## Any Base To Decimal



## Binary to Decimal



```

int ans=0;
int power=1;
while (n>0) {
    int lastdigit = n%10;
    ans = ans + power*lastdigit;
    n /= 10;
    power *= 2;
}
System.out.println(ans);
    
```

Variables:  
 $n = 101$   
 $ans = 0$   
 $power = 1 \cdot 2^0$

Loop Iterations:

- Iteration 1:  $n = 101$ ,  $power = 2^0 = 1$ ,  $ans = 0 + 1 \cdot 1 = 1$
- Iteration 2:  $n = 10$ ,  $power = 2^1 = 2$ ,  $ans = 1 + 2 \cdot 0 = 1$
- Iteration 3:  $n = 1$ ,  $power = 2^2 = 4$ ,  $ans = 1 + 4 \cdot 1 = 5$

$ld = 1$   
 $ans = 0 + 1 \cdot 1 = ans = 1$

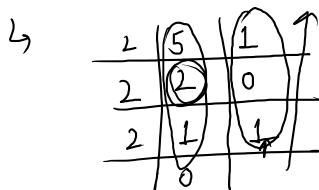
$ld = 0$   
 $ans = 1 + 2 \cdot 0 = 1$

$ld = 1$   
 $ans = 1 + 4 \cdot 1 = 5$

## Decimal to Any Base →

5 minutes ↗

## Decimal to Binary



$(101)_2$

```

int ans=0;
int power=1;
while(n>0) {
    int rem = n%b;
    ans = ans + power*rem;
    n = n/b;
    power *= 10;
}
System.out.println(ans);
    
```

Step out parenthesis (ans)'

```
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN,
     * Print results to STDOUT.
     * Your class name must be "Main"
     */
    int n = 5;
    String ans="";
    int power=1;
    int b=2;

    while(n>0){
        int rem = n%b;
        if(rem > 9){
            int dis = rem-10;
            ans=(char)(dis+'A')+ans;
        }else{
            ans=rem+ans;
        }
        n/=b;
        power*=10;
    }

    System.out.println(ans);
}
```