

Start

ARRAYS (Most Important)

Date _____
Page _____
Part - 1

why? → Whenever you want to store any data/any information we create variables. If we want more data we have to create more variables with different name. If we want to store 100 student roll number it's not possible to create 100 different variables. So, then ~~the~~ Array comes into picture.

e.g.: 100 Roll no →

```
int a; or int a,b,c,d,e, ... aa, ... aab, aac ...  
int b;  
int c;  
int d;  
int aab;
```

int roll = {1, 2, 3, 4, 5, ..., 99, 100};

what?

So, What is Array? ?

An array is a homogenous non-primitive data type used to save multiple elements (having the same data type) in a particular variable.

So, Arrays in java can hold primitive data types (Integers, Characters, Float etc) and Non-Primitive data type (Objects, Strings)

⇒ Array = i) Index based Data Structures to store large volume of Data using single variable name.

ii) Homogeneous data stores only

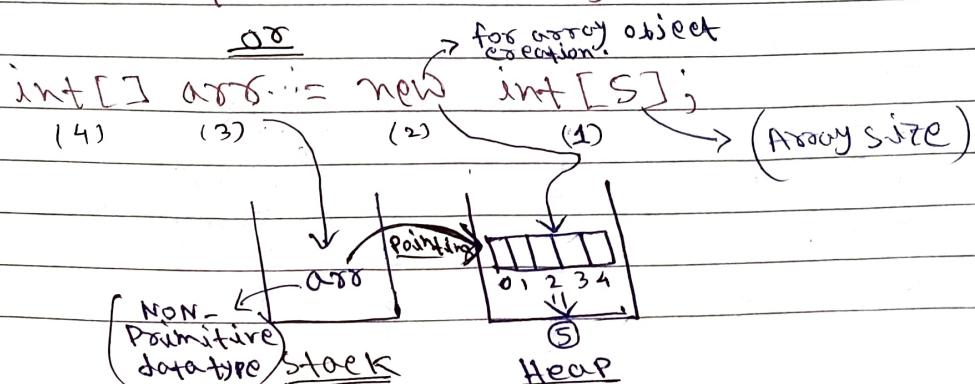
iii) Arrays in java is treated as objects → that's why the memory is allocated in Heap Area.

this is why we need new keyword to create array.

Short Definition: - Array is a index Based collection of same type of Data.

★ Syntax of array: —

```
int[] arr = {1, 2, 3, 4, 5, ...};
```



$\text{arr}[2] = 10;$ → [5 10]
 $\text{arr}[0] = 5;$

How?



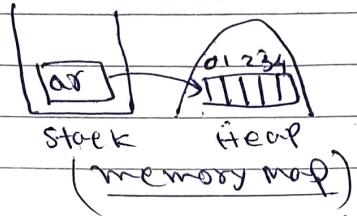
Cases:- (example)

1 D (Regular)
1 D (Jagged)
1 D (Nested)

① To store marks of 5 students →

→ Student $\Rightarrow \text{int}[\text{ }] \text{ or } \text{arr} = \text{new int}[5];$

1D Array



Syntax of 1D array

② To store marks of 3 classes each with 4 students →

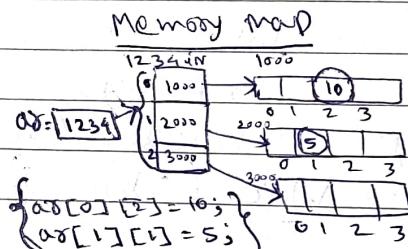
$\Rightarrow (\text{classes})(\text{marks}) \Rightarrow$

③ ④
 ↓ ↓
 0 4
 1 4
 2 4
 |
 (2D Regular Array)

Syntax of 2D = column row

$\text{int}[\text{ }][\text{ }] \text{ or } \text{arr} = \text{new int}[3][4];$

2D Array this is



NOTE:- these have two types of array in java.

a) type → ① Regular array, ② Jagged array.

b) Regular → Same type of data stored. e.g. [4]

b) Jagged → different or irregular type of

data stores. e.g. [5]
[3]
[4]

(3)

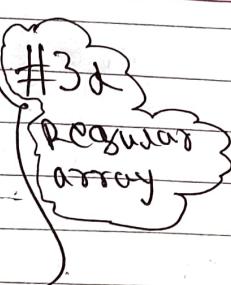
To store marks of 3 colleges, each college's 4 classes & class's 3 students each →

3D

(colleges)(classes)(students) \Rightarrow

Syntax 3D \Rightarrow

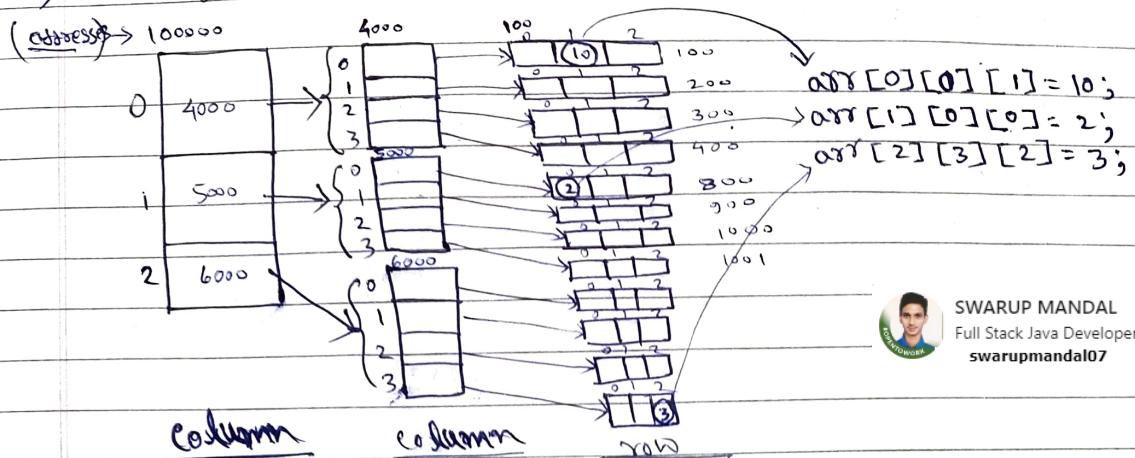
0	0	3
1	1	3
2	2	3
3	3	3
4	0	3
5	1	3
6	2	3
7	3	3



[3] [4] [3]
↓ ↓ ↓
Colour Colour Row

3D
Regular array

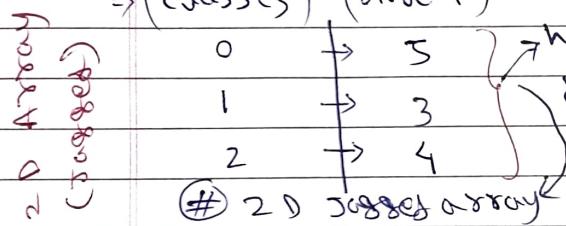
⇒ memory map of 3D Array;



SWARUP MANDAL
Full Stack Java Developer
[swarupmandal07](https://github.com/swarupmandal07)

(4) To store marks of 3 classes' with 5, 3, 4 Student respectively of classes: —

$\Rightarrow (\text{classes}), (\text{student})$



→ here, student of each classes' are different or irregular thus it is called as Jagged array.

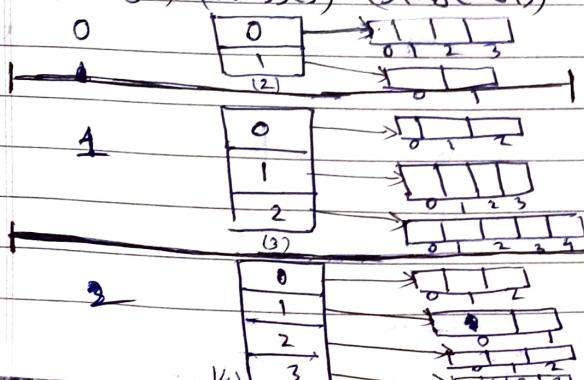
④ ⇒ Syntax of Jagged array :-

must give for 1D array
 do at least leave as blank
 (row)

int [] [] arr = new int [3] [] ;
arr [0] = new int [5] ;
arr [1] = new int [3] ;
arr [2] = new int [4] ;

⑤ To store marks of ③ colleges, each college has ②, 3, 4 classes respectively & class's [4,2] [3,4,5] [3,2,3,4] students →

⇒ (courses) (classes) (students)



Syntax of 3d Jagger:

`int [] [] [] arr = new int [3] [] []
declaration`

```

new arr[0] = new int[2][ ];
new arr[1] = new int[3][ ];
new arr[2] = new int[4][ ];

new arr[0][0] = new int[4];
new arr[0][1] = new int[2];

```

$\left\{ \begin{array}{l} arr[1][0] = \text{new int [3];} \\ arr[1][1] = \text{new int [4];} \\ arr[1][2] = \text{new int [5];} \\ arr[2][0] = \text{new int [3];} \\ arr[2][1] = \text{new int [2];} \\ arr[2][2] = \text{new int [3];} \\ arr[2][3] = \text{new int [4];} \end{array} \right.$

brackets (for different)

\Rightarrow if $\text{int}[\cdot][\cdot][\cdot].arr = \text{new int [3][2][?]}^3$

$\left\{ \begin{array}{l} arr[0][0] = \text{new int [3];} \\ arr[0][1] = \text{new int [4];} \\ arr[1][0] = \text{new int [2];} \\ arr[1][1] = \text{new int [3];} \\ arr[2][0] = \text{new int [3];} \\ arr[2][1] = \text{new int [4];} \end{array} \right.$

$3 \times 2 = 6$ rows

initialization

fill $arr[0][0][0] = 10;$

$arr[1][2][0] = 30;$

② Multi Dimensional Array:-

$\Rightarrow 1D, 2D, 3D \Rightarrow \text{Regular}$

$2D, 3D \Rightarrow \text{Jagged}$

4D

#Q&A

Q) $\text{while (1)} \{ // 1 is not boolean in java}$
snippets s.o.p("Hello");
 $\cdot \} \}$

Ans:- C. E

Exception Error

Q) $\text{while (true)} \Rightarrow$ it gives ~~infinite loop~~

Syntax:-
 while (boolean)
 $\quad // \text{stmt-1}$
 $\quad // \text{stmt-2}$
 $\}$ // without initialization
 and inc/decremnet while
 can run

so, it ~~continues~~ $\text{s.o.p("Swarnup");} // \text{single stmt-1 after while}$

\Rightarrow It gives infinite loop of continuously printing
 Swarnup as output. You have to forcefully stop it

Q) $\text{while (true)} \{$

$\}$ \Rightarrow It gives NO-output.

Q) $\text{while (true);} \Rightarrow$ It gives NO-output. ; is a ending of
 any statement

Q) $\text{while (true);} \Rightarrow$ It gives C. E

s.o.p("Swarnup");

Q) `while(true)` \Rightarrow It gives C.E (bcz declarative stmt not allowed in any loop) if you use without {} but in between {} it is valid.

Q) `while(true){` \Rightarrow NO OUTPUT
} $\int x = 10;$ \rightarrow to declaration.

Q) `while(true){` \Rightarrow it gives compilation error at line-n₂
} $S.0.\text{Print}("Hello") ; // \text{line-1}$ $\#(\text{unreachable code})$
 $S.0.\text{Print}("Hello"); // \text{line-2}$ \rightarrow This statement will continuously run (infinite)
}

Q) `while(false){` \Rightarrow it gives compilation error at line-n₁
} $S.0.\text{Print}("Hello") ; //$ if you make it false why you write stmt-
} $S.0.\text{Print}("Hello");$ $\#(\text{This expression will evaluate by JVM})$
}

Q) `int a=10, b=20;` \Rightarrow It gives nice as a infinite output
`while(a < b){` $\because (a < b = \text{true}) \Rightarrow 10 < 20 = \text{true}$
} $S.0.\text{Print}("Hello") ;$ \Rightarrow variable comes with value. and doing relational operation \Rightarrow result will be boolean. if true point (anything you want)

Q) `final a=10, b=20;` $\#(\text{This expression will evaluate by compiler})$
`while(a < b){` $\#(\text{Whenever variables are marked as final, compiler will try to know the value of those variables and it will use the values in the expression to get result})$
} $S.0.\text{Print}("a");$ $S.0.\text{Print}("b");$ $\#(\text{unreachable code})$ $\#(\text{line-2})$
} $\#(\text{line-1 give infinite loop})$

Q) `final a=10;` $\#(\text{evaluated by compiler (compiler knows)})$
`while(a < 20){` \Rightarrow It gives C.E at line #2
} $S.0.\text{Print}("a");$ $\#(\text{line-1})$
} $S.0.\text{Print}("b");$ $\#(\text{line-2})$

Q) `do` \Rightarrow do point the value and white(true) point infinite the statement of do.
} $S.0.\text{Print}("Hello") ; // \text{line-n}_1$
} $\#(\text{at least once})$
} white(true);

Q) `do;`
} white(true);
Ans:- NO OUTPUT

Q) `do`
} $\int x = 11; // \text{line-1}$
} white(true);
Ans:- C.E at line-1

Q) `do {`
} $\int x = 10; // \text{line-1}$
} white(true);
Ans:- NO output (✓)

P.S. VM (SE) 0.85

I. Q) do write(true); // P.E (because) while loop ~~with~~ inside
System.out.println("Hello"); // Stmt1 otherwise it will print
write(true);
} infinite the Stmt1

So, what compiler will do:-

do

write(true); ~~;~~ ~~;~~ remove

System.out.println("Hello");

write(true);

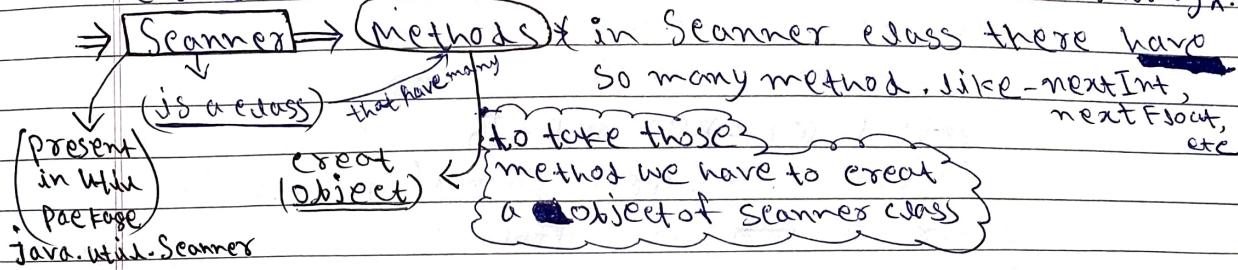


SWARUP MANDAL
Full Stack Java Developer
swarupmandal07

Part - 2

(*) User input from console in java:-

⇒ To take input from user we have to import a class called Scanner which is present in util library.



(*) Rules to taking input from User/console :-

- you have to import a package `import java.util.Scanner`
- To use the `Scanner` class's methods you have to create object of that class to give life and also pass a taking input Stream (`System.in`) which connect to the input

`Scanner scan = new Scanner(System.in);`

- Now, after creating object you can call all methods of `Scanner` class by the reference variable you created and store it in a local variable as per data type.

iii) a) To take integer input →

`int sm = scan.nextInt();`

↳ Autotype

↳ (L.V)

↳ (R.V)

↳ (method)

↓
input stream
System ⇒ final class
in ⇒ instance (variable)
out ⇒ instance of print stream
class

b) To take float input →

`float sm = scan.nextFloat();`

c) To take String input →

`String name = scan.next();`

to take input before space

`String name = scan.nextLine();`

to take input whose size increasing space

d) `nextDouble();` ⇒ to take double input.

e) `nextByte();` ⇒ " " byte " " .

f) `nextBoolean();` ⇒ " " Boolean " " .

g) `nextLong();` ⇒ " " Long " " .

h) `nextShort();` ⇒ " " Short " " .

i) `BigInteger sc = scan.next BigInteger();` ⇒ input as a BigInteger > long

j) `BigDecimal se = scan.next BigDecimal();` ⇒ take input as a BigDecimal > Double

k) `next().charAt(0);`
`nextLine().charAt(0);`

(import java.math.BigInteger)

(import java.math.BigDecimal)

> Double

1 D Array taking input (Regular)

→ Want to take marks from user and store it in array and want to display 5 student's marks.

```
import java.util.*;
```

```
public class Inputs {
```

```
    P.S.V.m(S[] a){
```

```
        int[] arr = new int[5];
```

```
        Scanner scan = new Scanner(System.in);
```

```
        // arr[0] = scan.nextInt();
```

```
        // arr[1] = scan.nextInt(); ----- till arr[4] (NO) X
```

```
        for(int i=0; i<5; i++){
```

```
            S.O.Pn("Enter marks of student: " + i);
```

```
            arr[i] = scan.nextInt();
```

taking
input
by for
loop

arr → [0 | 1 2 3] = 5
i = 0 & 5
arr[i] = 5
arr[0] = input

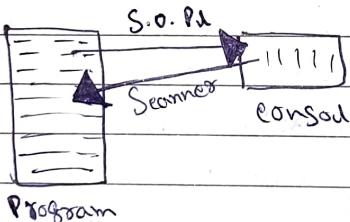
```
// Now want to display the taking input.
```

Display
input
by for
loop

```
{ for (int i=0; i<5; i++){
```

```
    S.O.Pn(arr[i] + " ");
```

```
}
```



Array last index = 6 ←
Array length = 6 + 1 = 7 ←
index a → 0 1 2 3 4 5 6

⇒ Length ⇒ it is a property of array. Length is a variable. Its duty is get the length of the array.

• Syntax:- $\text{Array Length} = (\text{Array's Last Index} + 1)$

array's variable_name.length

• ex:- arr.length;

```
⇒ int arr = new int[6];
```

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

```
    arr[i] = scan.nextInt();
```

```
}
```

⇒ Scanner scan = new Scanner(System.in);

```
S.O.Pn("Enter the number of student by user:");
```

```
int size = scan.nextInt();
```

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



classmate student

(j) (j)

innerloop
(j)
ordersoup(j)

2D Array taking input (Regular)

import java.util.*;

public class Input2d {

P.S.V.M(S> 0885{ (j) (j)}

int arr = new int [3][4]; // array declaration.

=> Scanner Scan = new Scanner (System.in);

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

for (int j=0; j<arr[i].length; j++) {

S.O.P("Enter marks of class "+i+" student "+j);

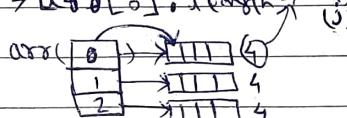
arr[i][j] = Scan.nextInt();

{ } { }

④ What happened?

∴ arr.length => i = 0; 0 < 3 => j = 0; 0 < 4
= [3] (i) (j)

∴ arr[i].length => arr[0].length => 4 (j)

arr[i][j] = Scan.nextInt();
arr[0][0] = 3

j++ => 0 to 1

arr[0][1] = 4

arr[0][2] = 5

arr[0][3] = 3

then again doing same

i++ => 0 > 1

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

for (int j=0; j<arr[i].length; j++) {

S.O.P("Marks of class "+i+" student "+j);

S.O.P(arr[i][j] + " ");

{ } { }

2D Jagged array taking input

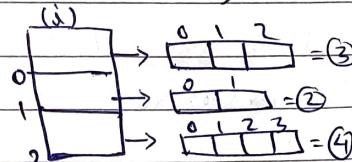
int arr = new int [3][];

arr[0] = new int [3];

arr[1] = new int [2];

arr[2] = new int [4];

= (3)

→ blank (How much know)
student didn't know

Scanner Scan = new Scanner (System.in);

for () // same for-loop for taking input (no change of previous)

for () // same for-loop for displaying input (previous)

arr[0].length = (3)

arr[1].length = (2)

arr[2].length = (4)

} Automatically
take

3D Regular Array taking input:-

colleges	class	students
(i)	(j)	(k)
0	0	3
	1	3
	2	3
1	0	3
	1	3
	2	3

```

⇒ import java.util.*;
public class Inputs {
    public static void main(String args) {
        Scanner sony = new Scanner(System.in);
        int[][][] arr = new int[2][3][3];
        #① for (int i=0; i<arr.length; i++) {
            ② for (int j=0; j<arr[i].length; j++) {
                ③ for (int k=0; k<arr[i][j].length; k++) {
                    System.out.println("Enter marks of colleges " + i + " class "
                        + j + " student" + k);
                    arr[i][j][k] = sony.nextInt();
                }
            }
        }
        System.out.println("Printing your typed input");
        # for (int i=0; i<arr.length; i++) {
            for (int j=0; j<arr[i].length; j++) {
                for (int k=0; k<arr[i][j].length; k++) {
                    System.out.println("marks of colleges " + i + " class" + j +
                        arr[i][j][k] + " " + "student" + k);
                }
            }
        }
    }
}

```

3D Jagged Array taking input:-

Scanner sony = new Scanner();

```

int[][][] arr = new int[2][][];
arr[0] = new int[2][];
arr[1] = new int[1][];
⇒ arr[0][0] = new int[3];
arr[0][1] = new int[4];
⇒ arr[1][0] = new int[5];

```

for (i) {

for (j) {

for (k) {

for (l) {

for (m) {

for (n) {

for (o) {

for (p) {

for (q) {

for (r) {

for (s) {

for (t) {

for (u) {

for (v) {

for (w) {

for (x) {

for (y) {

for (z) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

for (xx) {

for (yy) {

for (zz) {

for (aa) {

for (bb) {

for (cc) {

for (dd) {

for (ee) {

for (ff) {

for (gg) {

for (hh) {

for (ii) {

for (jj) {

for (kk) {

for (ll) {

for (mm) {

for (nn) {

for (oo) {

for (pp) {

for (qq) {

for (rr) {

for (ss) {

for (tt) {

for (uu) {

for (vv) {

for (ww) {

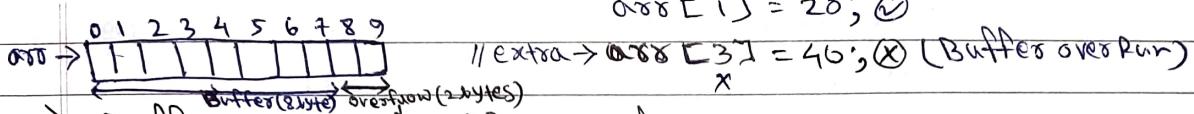
for (xx) {

for (yy) {

for (zz) {

Q) How Array data is stored in Ram?

⇒ Ram is a collection of bytes. So, when we create a array suppose (arr[2]) then we know it is 4 bytes. So, $2 \times 4 = 8$ bytes will be occupied in ram by those array. When we initialize our array by values of every indexes then all the bytes area will filled. If we filled extra indexes with value then in a Buffer overflow happened. $\rightarrow arr[2] \Rightarrow arr[0] = 10; @$
 $arr[1] = 20; @$



⇒ Buffer overrun, Buffer overflow:-

A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer which user chosen. It will be allowed for ~~user~~ write, it does not show any error. When we write it will show in compile-time. Buffer overflow can see in c, c++ programming.

⇒ Buffer overflow Attack?:-

Attackers exploit buffer overflow issues by overwriting the memory of an application. This changes the execution path of the program, triggering a response that damages files or exposes private information.

• Example:- an attacker may introduce extra code, sending new instructions to the application to gain access to IT systems. If attackers know the memory layout of a program, they can intentionally feed input that the buffer cannot store, and overwrite areas that hold executable code, replacing it with their own code. For example, an attacker can over write a pointer (an object that points to another area in memory) and point it to an exploit payload, to gain control over the program.

⇒ Type of Buffer overflow Attacks?:-

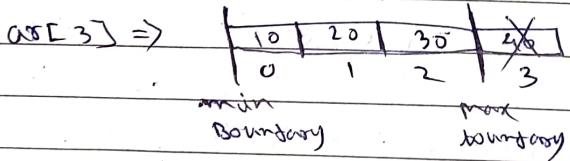
a) Stack-Based buffer overflow b) Heap-Based attacks

- ⇒ How to prevent Buffer overflows!—
- Developers can protect against buff. ove. via security measures in their code, or by using languages that offer built-in protection.
- ⇒ In addition, modern operating system have runtime protection. Three common protection are →
- a) address space randomization (ASLR)
 - b) Data Execution Prevention
 - c) Structured exception handler overwrite protection (SEHOP)

- ↳ d) The Improve Security solution!—
- i) DDoS protection, ii) web application firewall,
 - iii) Bot management, iv) Account Takeover protection,
 - v) API security vi) RASP, vii) Attack Analytics.

Q) Does in java Buffer overflow happens?

- ⇒ NO, Languages such as PERL, javascript, C++ and Java use built-in safety mechanisms that minimize the likelihood of buffer overflow.
- In other words, `data[array]` is bounded in java by Boundary → min. Boundary and max Boundary. If you write extra data for a location it gives Runtime error (Exception) ⇒ ArrayIndexes out of Bound. and all the code will terminated.



- ⇒ So, Array is Guarded with Boundaries to protect from Buffer overflow/run afteres.

- ⇒ We can't take input as a character only. But we can create character arrays no issue.
- ⇒ We can store or creat any type of array → int, char, float, double, long, String ...





Q) Creating a object's (including class) Array: -

class Fan {

```
    int cost; // instance variable / field / property
    String brandName;
    int noOfWings;
```

}

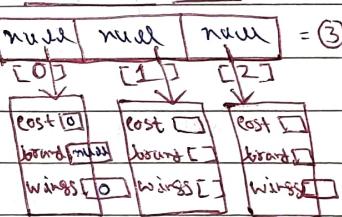
public class Array {

P. S. v. m (S[] arr){

Fan[] arr = new Fan[3]; // So, we can create array of
// creating Fan class array & objects every types/things)

{ arr[0] = new Fan(); } ← memory map
arr[1] = new Fan(); } → arr[0] ⇒ null null null = 3
arr[2] = new Fan(); } ↓ ↓ ↓
↓ ↓ ↓

memory map



{ arr[0].cost = "1000";

arr[0].brandName = "USHI";

('instances') (,) (,)

arr[0].noOfWings = 3;

arr[1].cost = 1100;

arr[1].brandName = "Havells";

? ?



SWARUP MANDAL

Full Stack Java Developer

[swarupmandal07](https://www.linkedin.com/in/swarupmandal07/)

⑦ Disadvantages of an array:-

i) It can store only homogenous type data.

`int a[] = new int [4];`

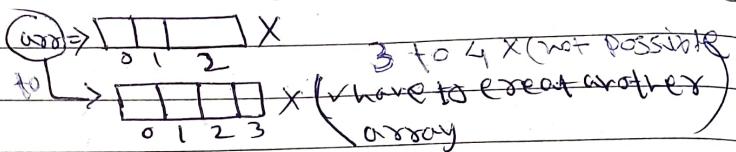
$a[0] = 10;$ ✓ 

$a[1] = 10.5f;$ ✗ \rightarrow only integer values

$a[2] = "inuron";$ ✗

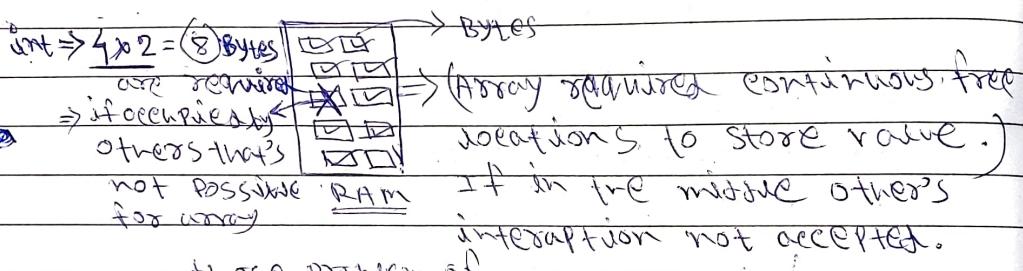
ii) memory of Array is fixed in size (it cannot grow or shrink)

`int a[] = new int [3];`



iii) Array demands contiguous memory locations:-

`int a[] = new int [2];`



So, To overcome array we have ArrayList.

⑧ => Recommended syntax of array:-

→ `int[] a = new int [2];`

You can use `[] a` or `a[]` does not matter
but recommended is `[] a`. Because it is more
readable by the user/developer.

→ `int a[] = new int [3];` // valid (But not Recommended)

means → `[a]` is 1D Array of int type data type

⑨ => We can directly assign value to the array

① 1D Array: `int [] a = {10, 20, 30, 40, 50};`

2 student each

② 2D Regular Array: - `int [][] a = {{10, 20}, {20, 30}};`

classes

(3) 2D Jagged :- $\text{int } [] [] a = \{ \{ 10, 20 \}, \{ 30, 40, 50 \} \}$

2 students of
0 class

3 students of
1 class

2 classes

4) 3D Regular :- $\text{int } [] [] [] a = \{ \{ \{ 10, 20 \}, \{ 20, 30 \} \}, \{ \{ 30, 40 \}, \{ 50, 60 \} \} \}$

→ traverses each & 2 student each

student

classes

5) 3D Jagged :- $\text{int } [] [] [] a = \{ \{ \{ 10, 20, 30 \}, \{ 10, 30 \} \}, \{ \{ 40, 50 \}, \{ 1, 2, 3, 4 \} \} \}$

3 student in classes 0 classes

(6) We can write :- $\text{int } [] a = \text{new int}[] \{ 10, 20, 30, 40 \};$
this is also possible for 1D, 2D, 3D array.

(Enhanced)

★ For-each loop :-

For-each is another array traversing technique like
for loop, while loop, do-while loop introduced in
Java 5.

⇒ Syntax :- $\text{for (type variable : array name)} \{ \text{Statement using variable;} \}$

⇒ example :- (1D Array)

$\text{int } [] a = \{ 10, 20, 30, 40 \};$

```
for (int element : a) {
    S.O. Pm(element);
}
```

$a \rightarrow \boxed{10 \ 20 \ 30 \ 40}$

go to first index

extract the value and give
to the variable (element)

then point it again
go to next index of

that array and keep
repeating.

⇒ when Whenever as a programmer, you are
not aware or you don't want to give
from which index you want to access
from which index you want to go. and

the program will go with the concept of For-each loop

⇒ In java for-each loop is used to iterate through elements
of arrays and collections (like linkedlist)

advantages :- i) it helps us avoid programming errors. ii) it makes the code precise
and readable. iii) easier to implement. iv) avoids the chance of
an infinite array.

disadvantages :- i) we can't jump over an element as it traverses through each elem.
ii) Traversing in reverse order not possible. iii) we can't modify array. iv) it's not
possible to keep track of the index. v) it has some performance overhead over