

DSA - kumar kushwah

Array

Searching

Sorting

String

Pattern | matrix

Recursion

Backtracking

bitwise , math

Linked list

Stack - Queue

trees

Heap , Trie , Graph

Sliding window

Dynamic Programming

Two pointer

HashMap

Interval

Divid & Conquer

Kadane's Algo

Nikita Types of Languages

Procedural Functional Object oriented

Java → follows object oriented & procedural
C → procedural

Python → functional + procedural + OO
C++ → OOA + procedural

Static vs Dynamic languages

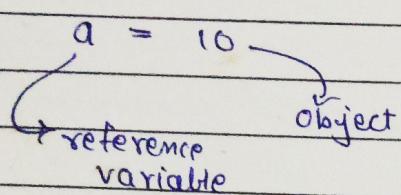
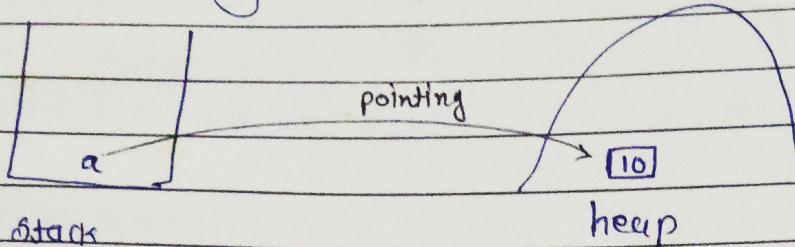
Static Dynamic

```
int a = 3;      a = 3;
int b = "Kunal" //error      b = "Kunal";
                                         //Compile time
```

Perform type checking at compile time Perform type checking at run time.

int a = 10;	error	when error a = 10;
a = "Kunal"		if next error a = "Kunal";
int a = 10;		
a = 60;		
		erro 'a' + 10

Memory management



one object

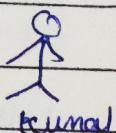
- more than one reference variables can point to the same object.
- if any one of these reference variables change the object, original object is going to be changed and it is going to be changed for all.

Friend

Son

brother

baby



if baby gets the hair cut
means kumar gets

then

Son, brother, friend

also have got the haircut

even though the change was made by one reference variable, original object was changed and since all of these other variables also pointing towards the original object. The change is going to be visible to all these variables.

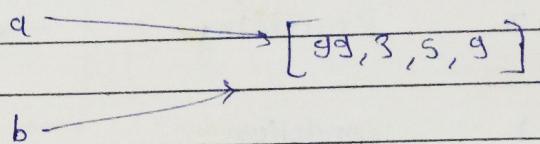
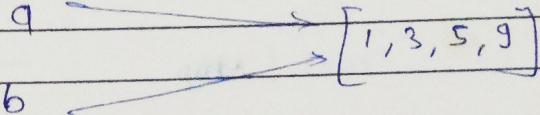
in java \rightarrow only

- pass by reference value.

$a = [1, 3, 5, 9]$

$b = a;$

$a[0] = 99$



output b

$[99, 3, 5, 9]$

Garbage collection

~~int a = 10;~~ $a \rightarrow 10$

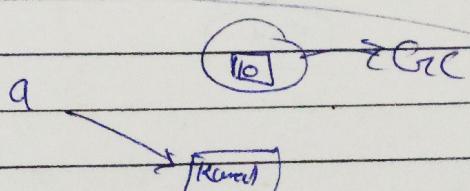
~~int a = 30;~~ $a \rightarrow 30$

it will took
all the objects
that do not have
reference variable.

in python

$a = 10;$

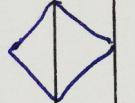
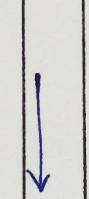
$a = "Kunal";$



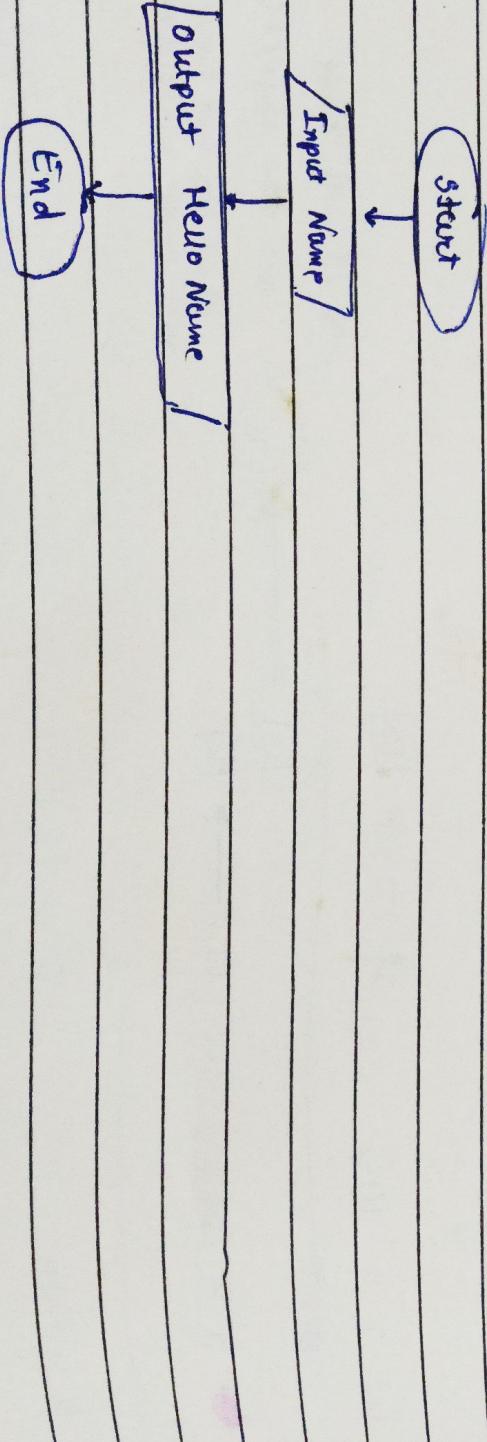
Day 2

Flow chart - Visualization of our thought process or algorithm and represent them diagrammatically is called flow chart.

Symbols to be used in flow chart

1. Start / stop → 
2. Input / output → 
3. Processing → 
4. Condition → 
5. Flow direction of program → 

Example 1 :- Take a name and output Hello name



Data Structure

data - Set

in Java notes

—Shrinath—
Date
Page No.

Example 2 Take input of a salary . if the salary is greater than 10,000 add bonus 2000 , otherwise add b. as 1000 .

Start

Pseudocode

[Input salary]

Start

input salary

if salary > 10000 :

 salary = salary + 2000

else:

 salary = salary + 1000

Output salary

exit

[Output salary]

end

- Pseudocode - Pseudocode is like a rough code which represents how the algorithm of a program works .
- Pseudocode does not require syntax

e.g. →

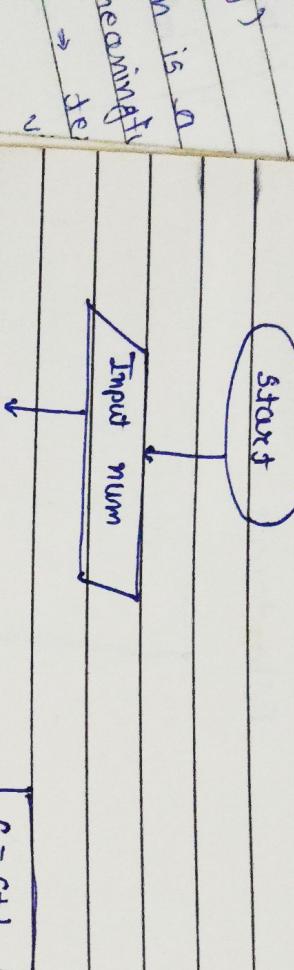
↓
 audio
 ↓
 video
 ↓
 mic
 ↓
 camera

for any

Shrinath
 Date _____
 Page No. _____

* * *

Example Input a number and



Pseudocode of example 3

```

start
input num
if num ≤ 1
  output "Neither prime nor composite"
else
  c = 2
  while (c < num)
    if num % c == 0
      output "Not prime"
    else
      c = c + 1
  end while
end
  
```

output "prime"

exit

```

print "Neither prime nor composite"
  
```

```

if num ≤ 1
  output "Neither prime nor composite"
  
```

```

else
  c = 2
  while (c < num)
    if num % c == 0
      output "Not prime"
    else
      c = c + 1
    end while
  end
  
```

```

  
```

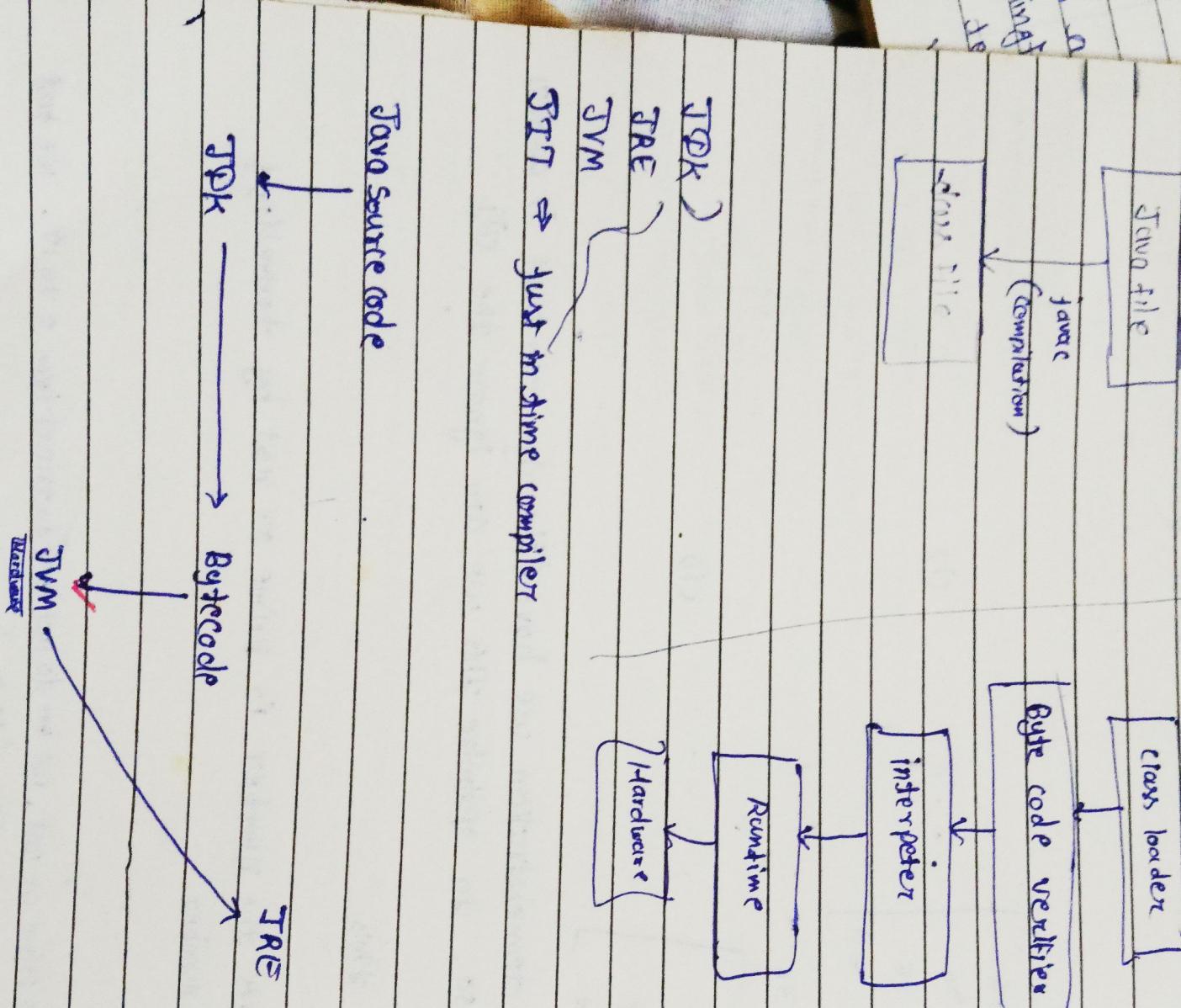
```

  
```

Java

Compile time

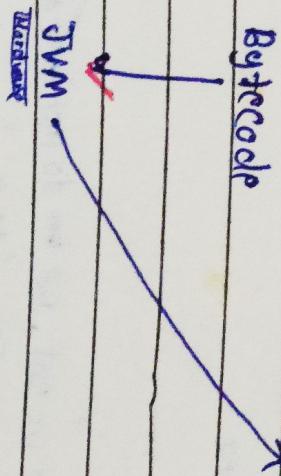
Run time



JIT \Rightarrow Just in time compiler

Java Source code

JRE



Day - 4H. Shrinath

Date

Page No.

Main.java

```
public class Main {
```

```
    public static void main ( String [ ] args )
```

```
    {
```

```
        System.out.println ("Hello world");
```

```
}
```

```
javac Main.java
```

```
Byte code
```

```
main.class
```

means

public \Rightarrow this class can be access from anywhere \rightarrow access modifier

class \Rightarrow group of properties and functions

name

main \Rightarrow file name (Main.java) and Main class is public and
main fn is also inside it

static \Rightarrow static variables and functions do not depend on objects

void \Rightarrow return type

public \Rightarrow so that everyone can access

(String [] args)

→ a array = collection of strings

Command line arguments

public class Demo {

public static void main (String [] args)

{ System.out.println (args[0] + args[1]); }

}

javac Demo.java

java Demo 30 "Kunal"

→ Command-line arguments

30kunal

← output

javac is a executable file

Command prompt

> where java c

> you will get the path remaining
we will cover later.

Arrays

// syntax

// datatype [] variableName = new datatype [size]

int [] rollnums = new int [5]

// or int [] rollnums = { 23, 42, 45, 36, 15 }

- datatype basically represents what is the type of data stored inside the array
- all the type of the data in the array should be same

// int [] rollnums ; // declaration of array

here rollnums is getting define in the stack.

// rollnums = new int [5]; // here object is being created in the memory (heap memory)
(initialization)

int [] arr = new int [5]

datatype

ref variable

creating the object
in heap memorythis thing
happens at compile timethis thing happens at
run timeRuntime
and known as dynamic
memory allocation

me)
audio
video
camera
mic
any

Shrinath
Date _____
Page No. _____

end way for each loop.

for (int num : arr) // for every element in array,
{ system.out.println (num + " "); } // print the element.
here num represents
element of the array

3rd way.

System.out.println (Arrays.toString (arr));

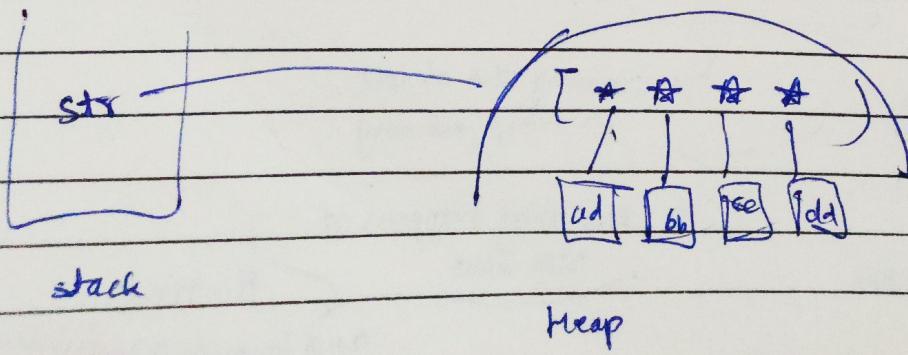
↑ Array of primitives ↑

Now Array of objects

String [] str = new String [4];

for (int i=0; i < str.length; i++)
str[i] = in.nextLine();

System.out.println (Arrays.toString (str));



array → mutable
String → immutable

Shrinath
Date
Page No.

20 array (Multidimensional arrays)

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

→ not mandatory

→ mandatory

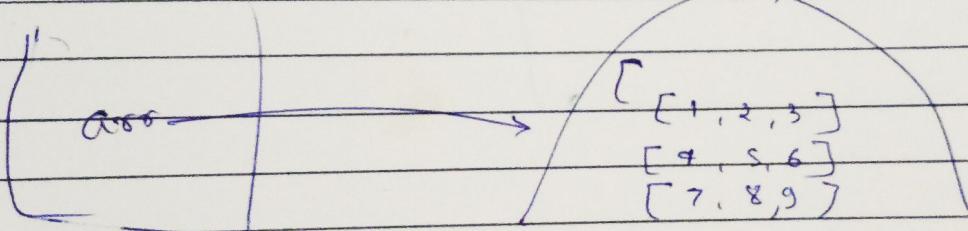
`int[][] arr = {`

{1, 2, 3}

{4, 5, 6}

{7, 8, 9}

`}`



imagine this as array of arrays

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

`int[][] arr 2D = {`

{1, 2, 3}, 11 0th index

{4, 5}, 11 1st index

{6, 7, 8, 9} 11 2nd index → arr[2] = {6, 7, 8, 9}

`}`

`for (row=0; row<3; row++)`

`{ for (column=0; column < size(arr[0]); column++)`

size of the row

arr[row].length

size (arr[0]);

`sop (arr[row][column] + " ");`

```
for ( int row = 0 ; row < arr.length ; row++ )  
{  
    System.out.println ( Arrays.toString ( arr [row] ) );  
}
```

enhanced for loop -

```
for ( int [] a : arr )
```

```
{  
    System.out.println ( Arrays.toString ( a ) );  
}
```

```
int [][] arr = {  
    { 1, 2, 3, 4 },  
    { 5, 6 },  
    { 7, 8, 9 }  
};
```

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

```
{  
    for ( int col = 0 ; col < arr [row].length ; col++ )  
    {  
        System.out.print ( arr [row] [col] + " " );  
    }  
}
```

```
S. C. p.m () ;
```

```
}
```

Linear Search

- Search in String
- Search in Range
- Minimum number
- Search in 2D Arrays
- Even digits
- Optimized for even digits
- Max wealth
- Outro

Search in string

boolean search (^{String} name, ^{char} target)

```
{ if (name.length() == 0)
    return false;
```

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

```
{ if (target == name.charAt(i))
    return true; }
```

```
return false;
```

for each

```
for (char ch : name.toCharArray())
```

```
{ if (ch == target)
```

```
    return true;
```

```
# System.out.println (Arrays.toString (name.toCharArray ()));
```

Search in 2D array.

PSV m(~~)

```
int[][] arr = { { 23, 4, 2 },  

                { 2, 3, 5 } },  

                { 19, 76, 12, 9 },  

                { 16, 12 } };
```

int target = 34;

```
int[] ans = search (arr, target);  

System.out.println ( Arrays.toString (ans) );
```

```
static int[] search (int[][] arr, int target)  

{  

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

        for (int col=0; col<arr[row].length; col++)  

            if (arr[row][col] == target)  

                return new int[]{row, col};  

}
```

~~return new~~ } return new int[] {-1, -1};

}

Maximum value in 2D Array.

leet - find no. of nos. that have even no. of digits

nums = [18, 129, 8, 1769, 98, 1]

ans = 3

```
psvm()
{
    sop ( numEven ( @.nums ) );
}
```

```
int numEven ( int [ ] nums )
{
    count = 0;
    for ( int num : nums )
        if ( evenis ( num ) )
            count++;
}
```

```
boolean evenis ( int digit )
{
    while ( digit > 0 )
    {
        c++;
        digit = digit / 10;
    }
    return c % 2 == 0;
}
```

if (digit < 0)
 digit = digit + -1;

if (digit == 0)
 c = 1;

we can replace it with $c = \lceil \text{int}(\text{Math.log10}(digit)) + 1 \rceil$;

Literals and identifiers →

int a = 10 ;

here 10 is a literal.

In primitive data types literals are the syntactic representation of boolean, character, numeric,

or here you can say string data

in int a = 10 ; the object is a literal

The reference variable a → is called the identifier

Identifier - name of the variable / method / class / packages
interfaces

in real life for readability

a = 234,000,000

in java

a = 234_000_000 ;

underscore will be ignored by compiler.

Output → 234000000

float marks = sc.nextFloat();
s.o.p (marks);

float marks = sc.nextFloat();
sop (marks);

input 134.92135678

input 10

output 134.92135

output 10.0

Floating point error

Type casting

Sum of two numbers.

`int a = sc.nextInt();`

if you input
a float value

10.35

it gives an error

But `float a = sc.nextFloat();`

and you
input a
int value

10

`System.out.println(a);`

then it's fine there is
no error

Output \Rightarrow 10.0

I was inputting an integer but it was automatically
converting it into float.

Typecasting

If when one type of data is assigned to another type
of variable then automatic type conversion will
take place, if the following conditions are met

- Condition 1 \Rightarrow the two types should be computable.

here float & int
but float & string

- Condition 2 \Rightarrow the destination type should be greater than the source
type.

here float > integer

Organization of data - Data structure (is used to store data)

Shrinath
Date
Page No.

if you want to assign a float value to an integer variable.

this will not happen automatically because integer is smaller than float

so this is sometimes called narrowing conversion

you are explicitly saying - "narrow the float into the integer"

```
int num = 67.56; // it gives error
```

```
int num = (int) (67.56 + );
```

→ `sop(num);` // it will give only the integer value of
67.56 means output 67

→ This is known as typecasting

compressing a bigger number in a smaller type explicitly

Automatic Type Promotion in Expression >

```
int a = 257;
byte b = (byte) a; // 257 % 256 = 1
```

\downarrow sop(b); } because the range of byte 256.
 \downarrow output = 1 it actually gives the remainder of 256 (max. value)

```
byte a = 90;
byte b = 50;
byte c = 100;
int d = a * b / c;
```

sop(d);

Output \Rightarrow 20

} $a * b \Rightarrow 2000$
 2000, not able to store in byte format
 $a * b$ easily exceeds the range of byte.

to handle this kind of problem
 java automatically is promoting
 each (byte / short / ...) to
 integer when its evaluating
 this expression ($a * b$)

```
byte b = 50;
b = b * 2;
```

\rightarrow this will give an error

byte evaluation are automatically converted
 $(b * 2)$ into int.

so this ($b * 2$) is an integer now
 so you can not assign it to byte.

Organization of data - Data structure (is used to store data)

Google → Unicode

ASCII

Shrinath

Date

Page No.

```
int number = 'a';
System.out.println(number);
```

} automatic typeconversion
it is giving us
ASCII value

Java follows the unicode principles

S.O.P ("नमस्ते");
Output → नमस्ते

Rules for type promotion -

- All the byte / short / character values are promoted to integer.
- If any one of the operands (that you are multiplying and dividing ...) is long then the whole operation will be promoted to long.

(operator) If it is float → then whole operation → float
double ⇒ entire operation → double

e.g. System.out.println(3 * 5.6);

System.out.println(3 * 5.6);
Output ⇒ 16.803709 Floating point error

Example of for thorough review of concepts.

byte b = 92;

char c = 'a';

short s = 1024;

int i = 50000;

float f = 5.67f;

double d = 0.1234;

double result = (f * b) + (i / c) - (d * s);

// float + int - double = double

S.O.P ((f * b) + " " + (i / c) + " " + (d * s));
S.O.P (result);

Watch it again and again casting is very important.

When you don't know how many times the loop is going to run, use the while loop.

When you know how many times the loop is going to run, use the for loop.

Organization of data → Data structure (is used to store data)

Shrinath

Date 8/4/22

Page No.

Day 5

- if else
- for loop
- while loop
- do while

find maximum value among three

int max = a;

```
if (b > max)
    max = b;
```

```
if (c > max)
    max = c;
```

```
s.o.p (max);
```

int max = Math.max(c, Math.max(a, b));

for taking a character input.

```
int ch = sc.nextLine().trim().charAt();
```

Day - 6

Java Switch Statement

quality content

we will study later ⇒ 26:11 min video

- Enhanced switch

DRY - Don't repeat yourself

Shrinath
Date _____
Page No. _____

Day - 7

Functions / Methods in Java

Function - Block of code that we can use it again and again

return-type name (arguments)

{ body }

return statement;

PSVM (~)

{

name = "Kunal";

name → "Kunal"

greet (name);

we pass value of the
the value means Kunal reference is passed
now

static void greet (~~String~~ naam)

{ s.o.p (naam); }

naam → "Kunal"

So finally

name → "Kunal"
naam → "Kunal"

in java there is only
pass by value. ⇐

pass by reference X

I. Data - Data structure (is used)

Shrinath
Date _____
Page No. _____

String name = "Kunal kshwah";

changeName(name);

static void changeName(String naam)

{ naam = " Rahul Rana"; } → naam → "RahulRan"

You cannot
use naam
outside
the method

S.O.P. (name)

name → "Kun. Kun."

}
psvm(~~)

int x = 10;

int y = 20;

swap(x,y);

" similarly & it will not
change swap

static void swap(int a, int b)

{ int temp = a;

a = b;

b = temp;

}

* primitives just passing value

* objects & stuff : passing value of the reference variable

non-primitive (pass by value)

Shrinath
Date _____
Page No. _____

```
public class ChangeValue {
```

```
    p. s. v. m. (String[] args)
```

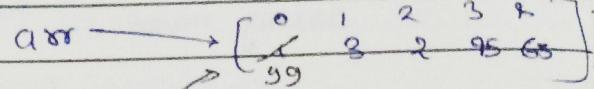
```
    { // create an array
```

```
        int[] arr = {1, 3, 2, 95, 6};
```

```
        change(arr);
```

```
        System.out.println(Arrays.toString(arr));
```

```
}
```



nums

nums[0] = 99;

original arr
will be changed

```
static void change (int[] nums)
```

```
{
```

```
    nums[0] = 99;
```

```
}
```

```
}
```

if you make a change to the
object via this ref. variable,
same object will be changed.

function's scope { }
loop Scoping

of data - Data structure (is used to store data)
 / Scop will begin when the value is initialized
 Date Shrinath
 Page No.

shadowing >

package com.kut

3 public class {

+ static int x = 90; // this will be shadowed at line 8

4 5 psvm (~r~)

6 { sop(x); // 90

7 int x = 40; // the class variable at line 4 is shadowed by this

8 sop(x); // 40

variable length argument -

import java.util.Arrays;

public class VarArgs {

psvm (String ~)

{ fun (... v: 2, 3, 4, 5, 56, 87, 23, 45, 65); }

fun();

}

Static void fun (int ...v)

{ System.out.println (Arrays.toString(v)); }

v1a

svf (String ...v)

}

Output []

}

audio

PSVM

```
multiple( 2, 3, "kunal", "Rahul", "Dhruv");
```

```
static void multiple( int a, int b, int ...v )
```

```
{ }
```

// rule varargs should be
always at length end

```
( int a, int...v , int b ) // error
```

- Function overloading

varargs can not be empty when you have done overriding

```
psvm } demo( ); } // error
```

```
static void demo( int ...v )  
{ }
```

```
static void demo( string ...v )  
{ }
```

Prim Number

Armstrong number

Armstrong all three digit

Data Structure

Data - ~~Set~~

In Java notes



Organization of data → Data structure (is used to organize)

→
Date
Page No.

Day - 8

```
String[] arr = new String[2];  
System.out.println(arr[0]);
```

Output ⇒ null

If it's a special literal, we can assign it only to non primitives

↳ String a = null;

int a = null × not possible CE

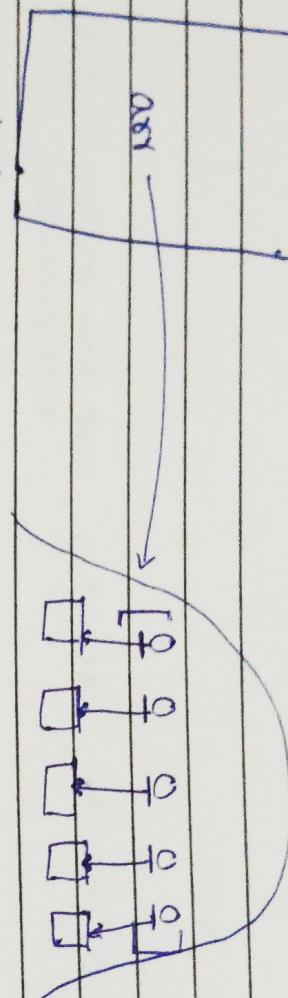
Any reference variable that you have by default it's going to have the null type if it is just a special value

primitives are stored in the stack memory (i.e. int, char)

But all the other objects like (string type, array type, user-defined type, hashmap) are stored in heap memory

String [] arr = new String [5];

// internal working of object



in heap memory \rightarrow array of string contains five elements

each particular element⁰ itself is an object, and each object will be stored in different part[□] of the memory

so this (0) is going be reference variable (something like arr[0])

since arr[0] is not initialized (hence we have not done arr[0] = "3";)

variable is point to null.

these reference variables (0) are pointing to objects that can be anywhere in the memory

So these reference variable (0), they may be like closely link together, when we talk about continuous memory allocation do you have continuous memory allocation? There is actually an array of reference variables

M. Shirinath
Date _____
Page No. _____

Array passing in functions

```
class {
    {
        public()
    }
}
```

```
int [] nums = { 3, 4, 5, 12 };
System.out.println(change(nums));
change(nums);
System.out.println(change(nums));
```

static void change (int[] arr)

```
{ arr[0] = 99;
}
```

```
nums → [ 3, 4, 5, 12 ]
```

```
arr → 99
```

Arrays are mutable in java



— M. Shrinath
Date _____
Page No. _____

Dynamic array

ArrayList

Ans :
why reason → Arrays are of fixed size
But if you don't know how much size you need.

class —

{
 psv m ()

} //syntax

ArrayList < Integer > lists = new ArrayList < > (10);
Or you can write —————— new ArrayList < Integer > ();

lists.add(57);

lists.add(99);

lists.add(139);

sop(lists);

sop(lists.contains(7593)); }
lists.set(0, 99)

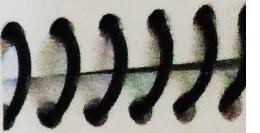
sop(lists);

List . remove ();
internally

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

 list.add(s.nextInt()); }

list.get();



Yash Shrinath
Date _____
Page No. _____

Multidimensional ArrayList

3

class _____

{
 public void _____

ArrayList < ArrayList < Integer >> lists = new ArrayList<ArrayList<Integer>>();

Initialization
for (int i=0; i<3; i++)
{
 lists.add(new ArrayList<>());
}

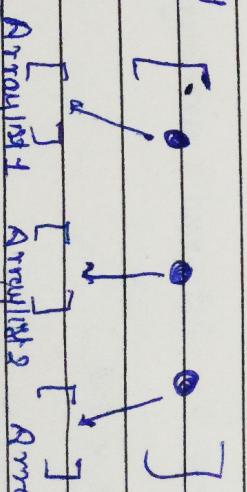
// add elements

for (int i=0; i<3; i++)

{
 for (int j=0; j<3; j++)
 {
 lists.get(i).add(sc.nextInt());
 }
}

scap(list);

internally
we can
assume



ArrayList1 ArrayList2 ArrayList3

Questions. Swap values, two elements of array
call swap(arr, ind1, ind2);

Question. Max item, can do it in the range

question max element

// work on edge cases

```
static int maxRange (int[] arr, int start, int end)
```

```
{  
    if (end > start)  
        return -1;  
    if (end == null)  
        return -1;
```

```
    int maxval = arr[start];
```

```
    for (int i = start; i <= end; i++)  
        if (arr[i] > maxval)  
            maxval = arr[i];  
    } }
```

Reversing an array

```
static void reverse (int[] arr)  
{  
    int start = 0; int end = arr.length - 1;  
    while (start < end)  
    {  
        // swap  
        swap (arr, start, end);  
        start++;  
    }  
    end--;
```

Organization of data - Data structure (is used to organize)



Day - 9 Linear Search

→ done before notes → before 22 page 3

Day - 10 Binary Search

- what is binary search
- Binary search Algorithm
- Why Binary search
- Code for Binary search
- Order agnostic binary search
- Code for order agnostic binary search

javaFolder ⇒ ~~#~~ OrderAgnosticBinarySearch.java

Day - 10 + hr. video Q. on Binary search

When to use binary search ⇒

1. in the problem statement, if you are given a sorted array.
2. you are required to get one particular answer and you are following a continuous sequence to get that answer. for e.g. square root of a number

public int length() { }

public int get(int index) { }

MountainArray and its methods

Given an interface

Same method definition \Leftarrow

So we will use
orderagnosticBinarySearch

- (Ans) Find peak element
- 1. if not found \rightarrow Linear search in descending array
 - 2. search in first half (as array) Simple binary search
 - 3. if not found \rightarrow Linear search in ascending array

arr [1, 2, 3, 4, 5, 3, 1]

1095 Find (searching) in a Mountain (Bidiagonal) array

using

Page No.
Date
Shrinath

Organisation of data \rightarrow data structure (if need of software)
In data notes

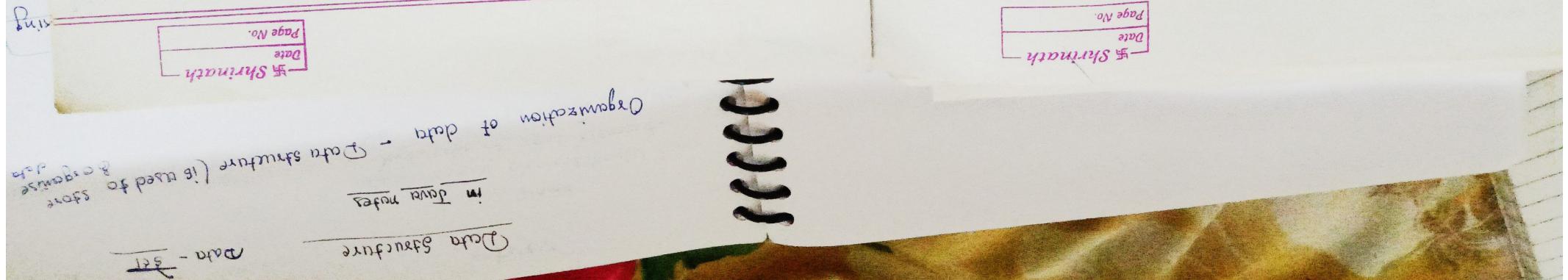
Data - 34

Data Structure

Shrinath

Page No.
Date

peak element in zig-zag array



1p Search in Rotated sorted Array

? Rotated array , not for duplicate elements

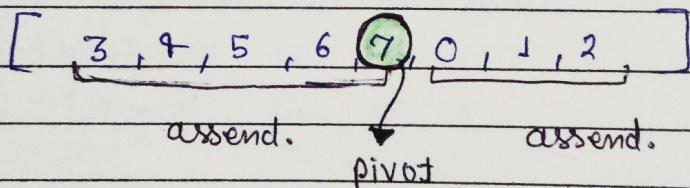
sorted array [2, 4, 5, 7, 8, 9, 10, 12]

after one rotation \rightarrow [12, 2, 4, 5, 7, 8, 9, 10]

after second rotation \rightarrow [10, 12, 2, 4, 5, 7, 8, 9]

1st approach \rightarrow find the pivot in the array

pivot \rightarrow from where your next nos are ascending



* Find pivot

* Search in first half \rightarrow simple Binary search (0, pivot)

* otherwise, search in second half \rightarrow simple B.S. (pivot+1, end)

ELG rotate the array.

nums [1 | 2 | 3 | 4 | 5 | 6 | 7] $K=3$ times

arr [5 | 6 | 7 | 1 | 2 | 3 | 4]

1 2 3 4 5 6 7
reverse reverse

4 3 2 1 7 6 5
reverse

[5 | 6 | 7 | 1 | 2 | 3 | 4]

Answer

if $n \leq k$
then $K = n$

$n = \text{nums.length}$, $k = \text{no. of rotation}$,

$n = 7$, $k = 3$

reverse(4, 6) ($n-k$, $n-1$)

reverse(0, $n-k-1$)

reverse(0, $n-1$)

reverse

[1, 2, 3, 4, 5]
 ^

Organization of data - Data structure (is used to store & organize)

Shrinath
Date
Page No.

YGM product of array except itself :-

pepcoding nuns [1, 2, 2, 3, 1]

nuns [12, 7, 6, 8, 12]

logic \Rightarrow do not use divide operator bcz there might be a element 0.

So we can do \Rightarrow prefix product from right
prefix product from left

nuns [1, 2, 2, 3, 1]

right [12 12 6 3 1]

left [1 2 * 12 12]

answer [i] = left[i-1] * right[i+1]

optimized

\rightarrow we can manage left with just a variable.
and create a array for right
 \Rightarrow watch soln at Leetcode