

3/10/22

Core Java

Hyder sir
Nitin sir

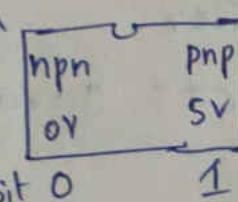
① How Computer works?

like about RAM

Mentors for this handwritten notes :

Hyder Abbas (linkedin.com/in/hyder-abbas-081820150/)
Nitin M (linkedin.com/in/nitin-m-110169136/)
Navin Reddy (linkedin.com/in/navinreddy20/) (Telusko)
written by pallada vijaykumar
[@ineuron.ai](http://ineuron.ai)

Technical discussion

1. CPU/microprocessor → 'Brain' of computer
* CPU consists of microprocessors and other components
* microprocessors consist of Transistors


Transistors understand only (0's & 1's)

0, 1 → machine level language.

* Instructions all at in one place called program to perform specific task

→ we have Machine level language (0, 1)

In MLL we can't write program to do task it's difficult to remember as a human's.

→ So, Then Assembly level language came into picture which consists of Alphabets and registers. Same this (ALL) also faces many issues

→ Then after High level language (HLL) came into picture, which consists of English like words and Simple Symbols like (+, -, *, %, /).

tx: MLL

0 1
1 0

tx: ALL

ADD
SUB
MUL
DIV

tx: HLL → (Human understandable)

+, -, *, %, /
print, scan like this
came in HLL

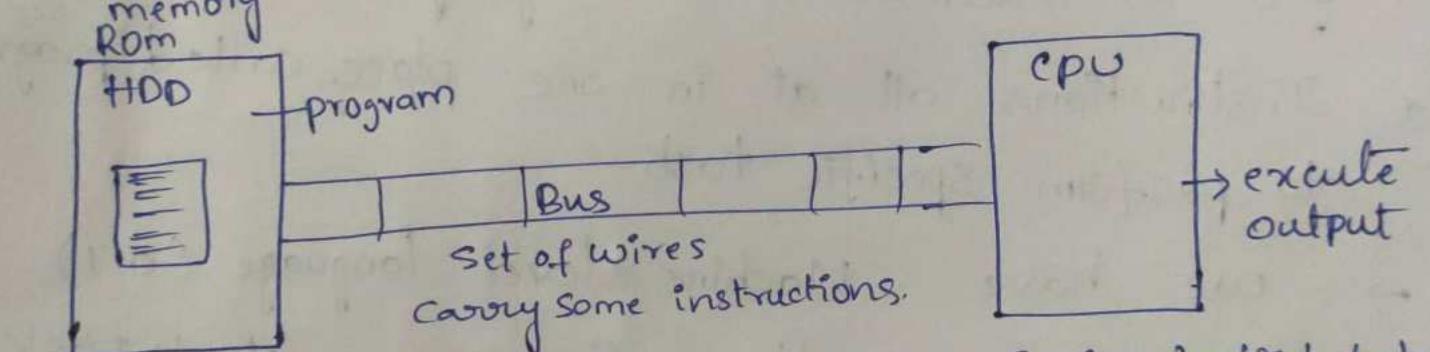
^(imp) Computer understands only machine level language which is 0's and 1's

So, to convert Assembly level language to machine level language we use Assembler
like mediator

and to convert High level language to machine level language we use "Compiler"

Examples for High level language : C, C++, Java, Python

② Inside Computer working

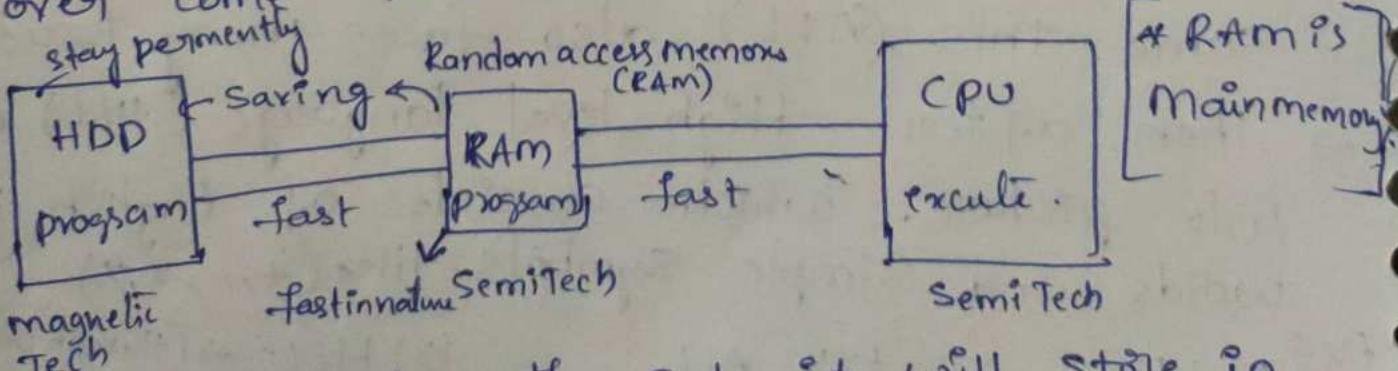


magnetic technology

Slow in nature.

so speed will mismatch b/w both

to over come this we RAM came into picture



* when we typing the code it will store in RAM only.

* Disadvantage of RAM is . it is volatile, continuous power supply needed.

* if power supply shut down during typing the code entire code will go be gone. again when we open. So to this we have to make "SAVE"(ctrl+s) everytime then it will store in HDD(Hardisk). In that harddisk program stays permanently until unless we delete.

But it will take time from deceiving program from HDD.

In HDD program storage we call Source file and in RAM we call it as Byte file. and in CPU we call it as Register.

* Cache memory closer to CPU. if we opening again again it will store in cash memory.

SSD → Solid State Disk Better Version of HDD
it use Semiconductor technology [flash storage technology]

③ (difference)
Object file (.obj file) and executable file (-exe)
.Obj is incomplete file
↓
complete file

* Source file → compiler → Obj file which is incomplete file.

we have add libraries → linking through linker.
and loading through loader → mix - Object file and libraries.

.exe is complete which includes all we execute to get output.

Technical discussion.

Java

C/C++ → Trending in 1990's

↳ 1991

↳ Sun microsystem → co founder Vinod Indian
↳ James Gosling

→ Easy to understand, object oriented, portable,
platform independent OR WORA

→ 1995 → Alpha & beta Version → trial Version
freely downloadable. / Open source.

→ Java 1.0 version released

Name : Oak, Green, Java

2011 - Oracle acquired Sun microsystem.

⇒ Object oriented.

→ platform independent / WORA

Internet programming language → when internet
came Java came.

Portable / platform independency / WORA

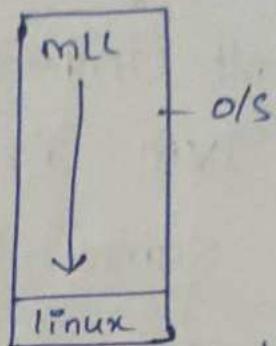
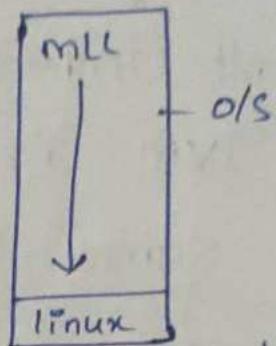
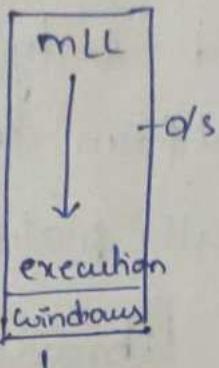
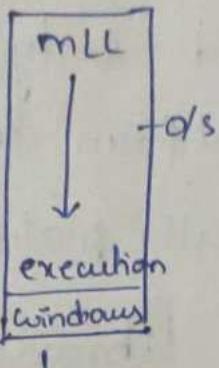
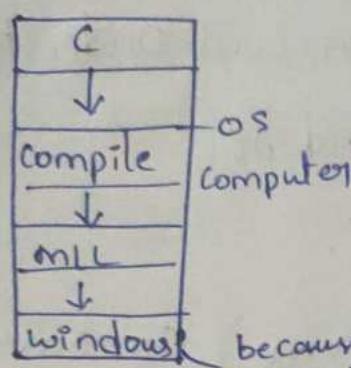
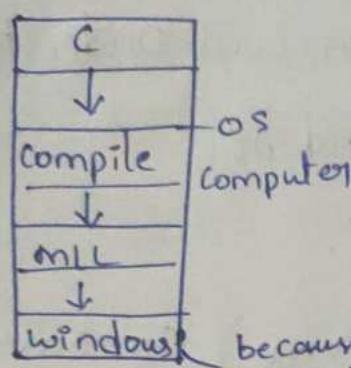
platform → microprocessor + Operating system

mp + OS

intel + windows

Software platform → Operating System.

platform dependency / Non-portable \rightarrow C is platform dependent because of its Architecture.



Result / O/P - Yes

Result / O/P - Yes

Program / App / Software \rightarrow Same

In C program if both are same platform.

MLL code will work gives you output.
because it is platform dependent.

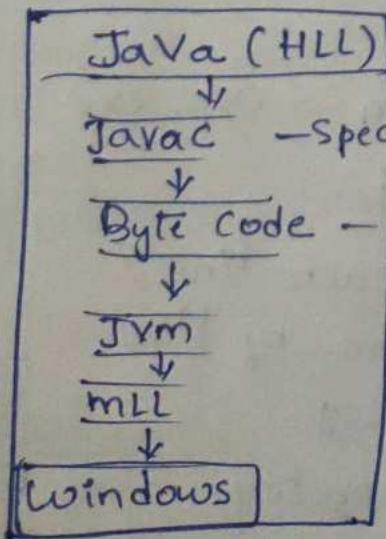
Actually code Never transports because anyone will copy

So coming to Java language.

Java \rightarrow platform independent \rightarrow (write Once run anywhere)

because it has special advantage

It will give you byte code called .class file after compiling .java file which is saved.



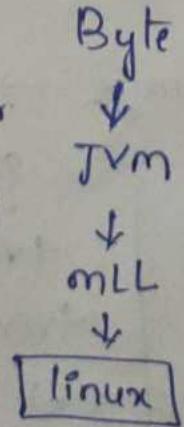
Java Compiler - Special Compiler

Byte Code - .class file

JVM

MLL

Windows



OS Platform

Byte Code

\downarrow

JVM

\downarrow

MLL

\downarrow

mac

OS Platform

Result \rightarrow O/P - Yes

We can share .class file (byte code) to any other platform we will get output.

.class file (byte code) is intermediate code human can't understand only JVM will understand because JVM is platform independent.

JVM is same in all platforms.

JDK → (Compiler, library, JVM) contains in JDK

JVM → Java virtual machine.

Java compiler → javac

Byte code (independent) like JVM / (MLL is dependent.)

(creation)
Java program

```
class Launch {  
    -  
    -  
}  
}
```

Source file

↓
javac → compiler → Byte code → JVM →
Launch.class

→ machine level language → platform → O/p. exe

Compiler

translate entire code at once

ex:- C / C++

Interpreter

translate each line /
translate line by line

ex:- python, J

Java is both Compiler & interpreter.

Famous Version of long term Support (LTS)

Java 7

Java 8 (in use now)

Java 11

Java 17

* why in C after compiling we will get MLL but not in Java.

Because C architecture is designed to get MLL and Java architecture is designed to get byte code

"C" is not to build internet applications

"Java" is to build internet applications.

means after Compile.

.class file → In this file byte code is there

.war file → collection of many .class file

.jar file → collection of many .class file.

.war file → web application archive

.jar → Java archive -

* what is purpose of giving arguments at the time of execution

Code → 3 phase before releasing the code to users

Developer → few inputs associated with developer

Testing → " " " Tester

Production → " " " , " , QA.

Editors → Typing code

IDE → Typing code, Compile, debug, Test
Integrated Development Environment (IDE)

Java is Verbose code → means we need to mention everything like class, main method.

→ What is method?

methods are like functions

we write methods to perform tasks.

method → name, parameters, Body, return type.

e.g.: $\text{Void } \underset{\substack{\text{return} \\ \uparrow}}{\text{name}} \text{ play}(\underset{\substack{\text{name} \\ \uparrow}}{\text{args}}) \{$
 $\}$ = -body

everything includes in curly braces.

→ Main method.

Starting point for program to start execution from there.

If we not have main method JVM will not start execution.

→ Why main method?

JVM has coded such way that JVM will start only if main method is there.

↑ to access without creation of object
public static void main (String [] args)
↓ ↓
TO make visible return typ
names it is starting point to execute program.

Return → forces the method to return value to the calling method of particular type execution of a method and can be used to return value from a method.

main → name of Jvm will search for this name
only to start execution of program void →
return type (Java main() will not return anything).

public → To increase visibility

Static → Can be accessed without object creation.

String args[] → To receive command line arguments.

Command line → From command prompt instructing something. arguments → information.

OS Javac launch .java

↓

Jvm → Java launch

Commandline
to execute

inewron

↳ argument

it will store here

class launch {

 public static void main (String args[]) {

{

 System.out.println("Hello");

 System.out.println(args[0]);

}

}

 } ↳ array
 inewron

 index 0

" " → whatever is in this print
That one

if we don't have " " it will print

"String" → we need to give String only Jvm is designed
inside that.
that way. And we pass only argument in command line
like "inewron" . 10 , 10.5 it will be converted into string.

Syntax's for Java :-

Public static void main (String [] args);
Public static void main (String args[]);
Public static public void main (String args[]);
Public static void main (String ... args);

↑ this can be any name.
} All valid.

Variable :- To store data

Static program language	Dynamical program
Java, C, C++	a=10;
int a=10; Before compile file we should specify what type of data.	type of data no need to mention

JDK → Compiler + JRE (Jit compiler) + library
JDK is required for developers as they write the code and run the code. JRE is required for end users as they only run the code.

1. Oop's (basic introduction).
2. Identifier | Variables. rules to write an identifier.
3. Reserved words.
4. Data types and its Chart.

Oops

it stands for object orientation principles.

Object: Real time instance of an entity.

every object in real time will have 2 parts

- ① what it has → having
- ② what it does → doing

e.g. Car,

what it has → name, noOfWheels, Speed.

what it does → move, brake, accelerate, will press
(Variable name ex:- noOfWheels)

class Car {

 String brandName;

 int noOfWheels;

[
.. classnames → it should
• start with capital letter
start each word with capital
ex:- NoOfWheels

} - statement ending

S.o.p("Hello");;);) → we can write how many;
Compiler will remove those;

Identifiers / Variables → it is a name in Java
Program.

it can be .classname, methodname, Variablename.

Rules for Writing an Identifier.

i) The only characters in Java identifiers

are → a to z, A to Z, 0 to 9, -(underline), \$

ii) if we use any other characters it would
result in . invalid. (Or) Compile time error.

- iii) Identifiers are not to start with digits
- iv) Java identifiers are case sensitive.
- v) There is no length limit on java identifier, but still it is a good practice to keep length of identifier not more than 15 character.
- vi) Reserved words can't use as identifiers.
- vii) pre defined classes are use as identifiers even though predefined class names can be used. but good practice we can't use.
- * Reserve words are normally in "lower case".

Reserve word

it is a built in words/keywords. which has already a predefined meaning to it 53 reserve words.

Data types

primitive data types.

- a. Numeric Value's
- b. Character Value's
- c. boolean Values.

Reserved words (53)

key words(50)

used key words 48

Unused keywords
(2)

↳ goto
const

literals

true
false } boolean
null

Literal \Rightarrow Any constant value which can be assigned to a variable.

`int data = 10;` \downarrow literal.

Data types : (Size and different values can be stored in variables)

Every variable has a data type.

Primitive data types : [that specify the type and size of data.]
a. Numeric Values
 \hookrightarrow to store number
primitive \hookrightarrow predefined by language and is named is reserved keyword.

i) integral data type (to store whole numbers)

① byte ($1\text{ byte} = 8\text{ bits}$)

Size of byte is $\rightarrow 8\text{ bits}$

min Value $\rightarrow -128$

[To prefer more space we can take (use) data type which has more space.]

max Value $\rightarrow 127$

byte marks = 35; Valid

byte marks = 135; Invalid

② short = 16 bits (2 bytes) [When we perform the operation compiler is designed to store data in higher data type of specific type.]
min Value = -32768
max Value = 32767

③ int = 32 bits (4 bytes) [Java give preference to int data type as it has higher precedence order.
min Value = -2147483648
max Value = 2147483647
 \rightarrow it is commonly used.]

By default if we specify any literal of number type compiler will try to keep it as "int" only. But we can also keep in short (or) byte.

4) long = 64 bits (8 bytes)

minValue = -9223372036854775808

maxValue = 9223372036854775807.

when we work with large files, data would come to Java program in terms of GB's.

when int is not enough to hold bigger values then we can use long data type (long data = 123456789101112L;)

By default compiler will treat like integer but if we need to store in long data type at last of literals we should mention with capital "L"; (or) "l" Small.

i) To store real numbers

as float

Size = 32 bits (4 bytes) (why it will treat double)

min Value = 1.4E-45

max Value = 3.4028235E38

Eg:

float number = 10.5f;

By default we will specify any real number compiler treat as double. To specify to compiler to treat it as float suffix it with 'F' (or) 'f'.

double

Size = 64 Bits (8 bytes)

Min Value = 4.9 E - 324

Max Value = 1.7976931348623157E308

Eg: double f = 10.5d;
(or)

double f = 10.5;

To map primitive data as Object in Java from
JDK 1.5 Concept of Wrapper Class.

Data types → follows specific format to store in
0's & 1's
To perform operations.

follows based on

byte = Byte (class)
short = Short (class)
double = Double (class)
long = Long (class)
int = Integer (class)
float = Float (class)
char = Character (class)

Now char : To store any character or special
character (Team IEEE \Rightarrow 65536 character are there).

Size = 2 Byte (16 bits) \Rightarrow It follows UTF(Unicode).

Syntax : char a = 'A';

Java is not pure OOPS because we use primitive data types.

By using wrapper class we can make 100% OOPS.

Ex :- int a = 25;

int b = 2;

Output :- 12,

float c = a/b f ; Truncation / Rounding to zero
s.o.p(c);

If we divide int by int output also int
truncation means if we are storing int in int the
value output (also) stored in int

Type casting :- changing the data from one
data type to another data type.

Two types 1) implicit conversion
2) explicit conversion

1) implicit :- means internally changing
by Java compiler from lower data to higher
data type

byte a = 10; Output :- 10

double b;

b = a; ~~int sh~~

s.o.p(b);

byte short int long float double

→ → → → →
implicit (small to large)

Explicit : It is done by programmer If convert larger data type to smaller data type.

Ex:- int a=500;

byte b;

a=(byte)b

we are using explicit word

s.o.p(b);

Output:- -15 loss of precision (or) data will happen.

Operators

operator : To do any operation.

$a+b = 10$ - literal
operator
operands

Incrementation

To increment value
(or)

Increase existing value
by 1

$(a = a + 1)$ we can write

it as $a++$, $++a$

↓ ↓

post increment preincrement

Decrementation

To decrement value
(or)

decrease existing by 1

$(a = a - 1)$ we can write it as

$a--$, $--a$

↓ ↓

post decrement pre decrement.

Post increment

```
int a=5;
```

```
int b;
```

```
b = a++;
```

a 6

b 5

```
S.O.P(a);
```

```
S.O.P(b);
```

Output: 6 5

Note: post & pre increment gives u same value after increment

But when we are going to assign value another variable then it will change

Ex: Post increment.

```
int a=5;
```

```
int b;
```

Output: a 6
b 5

```
b = a++;
```

In this first use value of 'a' and assign to 'b' then increment in 'a'

Ex: Pre-increment

```
int a=5;
```

```
int b;
```

```
b = a++a;
```

In this first increment value of 'a' then that & value assign to 'b'

Output: a 6
b 6

decrement post

```
int a=5
```

```
int b;
```

```
b = a--;
```

Note: post & pre decrement gives u same value after decrement we are going to assign value another variable then it will change.

Ex: post decrement

```
int a=5;
```

```
int b;
```

```
b = a--;
```

a 4
b 5

```
S.O.P(b);
```

In this first use value of 'a' and assign to 'b' then decrement in 'a'

Ex: pre-decrement

```
int a=5;
```

```
int b;
```

```
b = --a;
```

In this first decrement value of 'a' then that 'a' value assign to 'b'

Output a 4
b 4

Ex: `int a=5; a` 5678

`int b;` `b` 20
`b= a++ + ++a + ++a` 5 6 7 8

`S.o.p(a);`

`S.o.p(b);`

output
58
20

Ex: `int a=5;`

`int b;` 6 7 8 9
`b= ++a + a++ + ++a + a--;` 5 6 7 8

`S.o.p(a);` `a` 56787 `7`

`S.o.p(b);` `b` 28

Comments → JVM will not execute
for single line comment (//)
for double line comment (/*-----*/)

Mentors for this handwritten notes :

Hyder Abbas (linkedin.com/in/hyder-abbas-081820150/)

Nitin M (linkedin.com/in/nitin-m-110169136/)

Navin Reddy (linkedin.com/in/navinreddy20/) (Telusko)

written by pallada vijaykumar

@ineuron.ai

Operators → fundamentals of Java.

Arthematic operations

+ , - , * , / , (%)
+ , - , * , / , (%)
add sub mul div ^{quotient}
add sub mul div Module (Remainder)

add sub mul div Module (Remainder)

This are normal very simple operations

```
int a=10;  
int b=20;  
int c=a+b; //30  
int d=a-b; //-10  
int e=a*b; //200  
int f=a/b; //1  
int g=a%b; //0
```

$$\begin{array}{r} 10) 20 \text{ (2 - quotient} \\ \underline{-20} \\ \hline 0 \text{ rem} \end{array}$$

Logical operator

&& , || , ! - NOT
↓ |
AND OR

0	→ False
1	→ True

NOT (Opposite)

0 → True
1 → False

AND

00 - false
11 - true
01 - false
10 - false

OR

00 - false
11 - true
01 - true
10 - true

AND → if all are true then it will true if one is false then it will false as result

OR → All expressions are true then it will true and if one false is there then it will true

All are false then it will false.

NOT : if quite opposite if expression is true then it will false. if expression is false then it will true.

Relational operator

< , > , \geq , \leq , == , !=

Comparison , Relate \rightarrow we can do this things

a = 10;

b = 20; and if two are false it will give false
a \geq b; \rightarrow In this it will check one is true it will true

a \leq b; \rightarrow If one condition is satisfy it will give true and if two conditions are not satisfy it will give false.

a != b $\begin{matrix} \text{yes.} \\ \xrightarrow{\text{No - same}} \end{matrix}$ it will give true.

To compare two values we are using '=='

ex: $10 \stackrel{\uparrow}{=} 10;$

To assign a value to Variable . we will use " = "

ex: $a \stackrel{\uparrow}{=} 10;$

Assignment Operators:

(==) \rightarrow single assignment

L) to assign value to Variable . we are using

\leftarrow ex: $a \leftarrow 10;$ Right to left.

chained assignment

int a, b, c, d;

$\underbrace{a=b=c=d}_{4} = 10;$

chained assignment.

a [10] , b [10] , c [10] , d [10].

If all are variables in same data type then
only we can do chained assignment.

If all are variables in same data type then
only we can do chained assignment.

int abc = 5; Variable can hold one value

int abc = 10; at a time, if we assign
new value then old value will
go off.

Compound assign (or) shorthand assign

int a=10;

a+=20; \rightarrow $a=a+20 \Rightarrow a=10+20 \Rightarrow 30$

a*=20; \rightarrow $a=a*20 \Rightarrow 10*20 \Rightarrow 200$

a-=20; \rightarrow $a=a-20 \Rightarrow 10-20 \Rightarrow -10$.

a/=20; \rightarrow $a=a/20 \Rightarrow 10/20 \Rightarrow \cancel{0}2$

a% = 20; \rightarrow $a \% 20 \Rightarrow 10 \% 20 \Rightarrow 0$

a++ is different

$a = a + 1 \rightarrow$ it will increase by only 1 value

Unary :

one operand is enough to do operations.

Ex: int a=10

⁽¹⁾
 |
 | One operand

Assign operator.

Ex: Assignment

& increment

decrement.

binary

more than one operand is involved to do operation

→ Relational logical

Arithmetic operation

$a=10;$
 $b=20;$
 $a+b \rightarrow$ Both

operations need.

Conditional :-

if we are doing any task based on condition.

int $a=10;$

int $b=5;$

if - else

(only one condition)

{ }

(curly braces)

if ($a>b$) {

int res = $a-b;$

s.o.p(res);

}

else

{ int res = $a+b;$

s.o.p(res);

}

[out put = 5]

if (condition) (if true)

{

body - execute

}

(or) if false

else {

body

- else block

}

else if (more than one condition)

int $a=10;$

int $b=2;$

if ($a>b$)

{

s.o.p($a+b$);

}

else if ($a=b$) {

s.o.p($a+b$);

}

else { s.o.p($a*b$); }

we can write one more
"else if" in "if else"
how much we can't
want conditions.

Nested - if :- in one 'if' another 'if' condition

int a = 10;

int b = 2;

if (a > b) {

 if (a == 10) {

 s.o.p(a - b);

 }

 else {

 s.o.p("Not doing");

}

}

if (condition) {
 s.o.p("*");

}

else if (condition) {
 body ;

}

else {

 s.o.p("*)");

}

in if else we wrote
another if condition 'elseif'

By using logical operator & Conditional operator

int a = 10;

int b = 20;

int c = 30;

if (a < b $\&&$ a < c)

{
 s.o.p("A is least");

}

else if (b < a $\&&$ b < c);

{
 s.o.p("b is high");

}

else {
 s.o.p("c is high");

}

if both are true only
it will true it will
execute

execute and check
(practice)

Ternary question mark if condition checking
(condition) ? T : F ;

↳ if condition is true before colon (:) it will execute. (T part will execute)

if condition is False After colon (:) it will execute (F part will execute)

Ex: ① `int a = 10;`
`int b = 20;` if false so After colon will execute
`int c = (a < b) ? a : b` Output
 [20]
 so. Ans is b
`System.out.println(c);`

Ex: ② `int a = 10;` before colon will execute.
`int b = 20;` so
`int c = 30;` true before colon
`int yes = (a < b) ? (a < c) ? a : c : b & c ? b : c ;` After colon

(variable name)

to

store result

| in before colon also we have another Ternary operation

$(a < c) ? a : c$

$(10 < 30) ? 10 : 30$

True

So before colon will execute

Ans [10]

Switch Case

Switch (Condition) {

↳ it will match with below case that will execute.

Case 1;
S.o.p();
Case 2
S.o.p();

If case 100 last outcome

{ }
Ex:-

int number = 100;

Switch (number) {

just for reference

this is match it will print control go out of switch

Last outcome
An output
arrow

① Case 100 :

System.out.println("not equal to 100");

② Case 50:

System.out.println("not equal to 100");

③ Case 10:

System.out.println("not equal to 100");

In case 100 case 100 is in ③ it will check one case it will execute ~~that case~~ until case is equal to 100. So. to avoid this we have to write (break;) after each case

At last we can write default :

System.out.println("message");

If none of the case is not match default case will execute.

Once check in IDE code in all possible ways.

Just for knowing purpose. formulas

① Percentage total obtained by out of marks 3 students
float percentage = $\frac{\text{total} * 100}{300}$;

Swapping without third Variable

$$b = b - a;$$

$$a = a + b;$$

$$b = - (b - a);$$

$$a = -6$$

$$b = 8$$

$$b = 8 - (-6)$$

$$b = 14$$

$$a = -6 + 14$$

$$a = 8$$

$$b = -(4 - 8)$$

$$b = -6$$

② $\text{Fahrenheit} = (\text{Celsius} * 1.8) + 32$.

leap year $\Rightarrow \frac{n}{4} == 0$ & $n \% 400 == 0$
no. of year

③ To print ASCII Value of char

s.o.p((int)'h');

④ char a = 'A' ***

int b;

b = a;

it will print ASCII Value of 'A'

when we are assing ~~integer-type~~
char type to integer

It will give u int value only

Loops

System.out.println(); After printing it will go next line. tell Vijay I - cursor went line

System.out.print(); After printing it will remain in same tell Vijay I - cursor same line

Loops : To Repeat any activity (or) Same activity to repeat. 3 types 1) for
2) while
3) do while

For loop

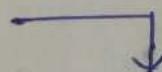
Three things are manditory to execute for loop like initialization, condition, update).

Initialization ①

(4 steps)



Condition check (No)



↓ (Yes) ②

(Stop)

Body of loop ③

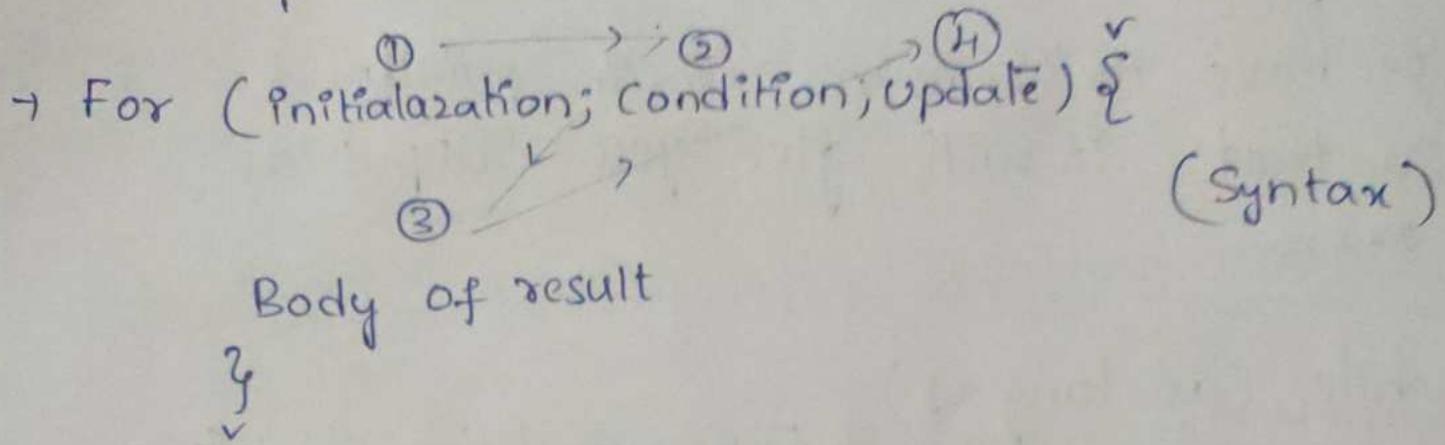


Update ④ (increment or decrement)

* until unless condition is true it will execute Body of loop

and keep on update the value and condition check. Condition fails it will stop.

To print 5 (*) stars



Ex:- `for (int i=0; i<5; i++) {
 System.out.println("*");
}`

Output

0 *
1 *
2 *
3 *
4 *

→ ① If i is starting from (0 to 4) so 5 numbers only

0 1 2 3 4
 5

* int n=5. we are using
* For (int i=0; i<n; i++)
* {
* System.out.println("*");
* }
we can write n here
instead of indirect number
we have to initialize n
first already

if you want to change number of stars
Just in above initialization of 'n' we can
make replace 5 with 9

Ex:- `int n=5;
(Start with 0) int n=9;
for (int i=0; i<n; i++)
{
 System.out.println("*");
}`

Output: it will give

u 10 stars

0

1

2

3

4

5

6

7

8

9

total 10 →

*
*
*
*
*
*
*
*
*
*

+ print that start horizontal
we have to remove \n-in println()
so that it will give you output as
but * * * * *

while (as long as)

int n=5; * Entry condition check loop
int i=0; ① * we can execute based on
while (i < n) { ② condition no need of update
 System.out.println(③ "*"); or initialization like (for)
 i++; ④ } if you want output as
 * * * * * like this
 remove \n in println()
 in program.

here first initialize value
and (as long as) while (i < n) checking condition
if condition is OK after control will go to body
execute and update ① (checkin)
i = 0 ② i < n
n = 5 ③ 0 < 5 - true - so print (*)
 like it will continue
 once 5 < 5 - fails so nothing print
 terminates loop

do while (exit condition check loop)

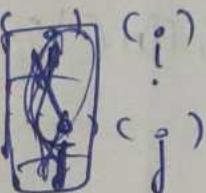
```
int n=5  
int i=0  
do {  
    System.out.println("*");  
    i++;  
} while (i < n);
```

irrespective of condition
atleast once it will execute
for second time to repeat
it will check condition
But first time it will execute
without condition

Unlike for, while, only first condition requires here do-while - no condition required first time. for second time its required condition

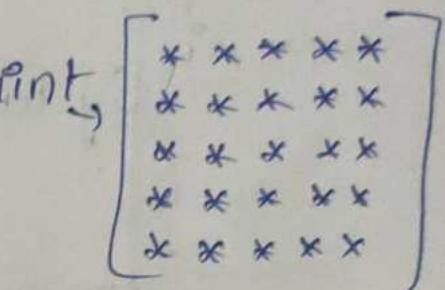
nested loop :- means inside loop one more loop we can write.

Ex:- write a program to print,

So treat rows as 

$n=5$ - no. of rows

$i \rightarrow$ $\downarrow j$
 $j=0 \quad j=1 \quad j=2 \quad j=3 \quad j=4$
 $i=0 \quad * \quad * \quad x \quad x \quad x \quad x$
 $i=1 \quad * \quad * \quad x \quad x \quad x \quad x$
 $i=2 \quad * \quad * \quad * \quad x \quad x \quad x$
 $i=3 \quad * \quad * \quad * \quad * \quad x \quad x$
 $i=4 \quad * \quad * \quad * \quad x \quad x \quad x$



rows (i)
↓
columns (j)

we are starting here from '0' so ~~to~~(0 to 4)

program :- ①

int n=5;

```
for (int i=0; i<n; i++) {  
    for (int j=0; j<n; j++) {  
        System.out.print("*");  
    }  
}
```

output:



System.out.println(); — to change cursor to next line.

Program (1) Break down & Debugging

```
int n=5;  
for ( int i=0; i<n; i++ )  
{  
    for ( int j=0; j<n; j++ ) → i=0 * * * * *  
    {  
        System.out.print ("*"); i=1 * * * * *  
    } i=2 * * * * *  
    System.out.println (); i=3 * * * * *  
}  
}
```

→ * first we are initialize n value to 5 (0 to 4)
we are starting from '0'

* i 0 → $0 < 5 = \text{true}$ so control will go into body
in the body we have another loop 'j'
J 0 → $0 < 5 = \text{true}$. so control will go into body
in the body we have printing '*' After
printing '*' j value is going to update → After this output
same above will process

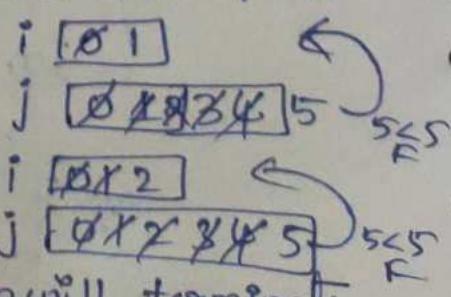
j 1 → $1 < 5$ - same above will process
until unless j value to 4 loop will execute
when j value 5 ($5 < 5$) false. so terminate.
and again control goto 0 & update i value

to i 1 $1 < 5$ - true so again above will take place

Same for i=2

i=3 same process
i=4

when i=5 $5 < 5$ - false loop will terminate
everything will stop



Program ③

Write a program to print pattern

\downarrow
 i=0 j=0 j=1 j=2 j=3 j=4
 * * * * * *

i=1 * - - - *
 i=2 * - - - *
 i=3 * - - - *
 i=4 * * * * *

$$\therefore i=4 \quad (i) = (n-1)$$

$$\begin{aligned} &= n-1 \\ &= 5-1 \\ &= 4 \end{aligned}$$

So. both are same

$$\frac{n-1}{2} = \frac{5-1}{2} = \frac{4}{2}$$

$$\boxed{\frac{n-1}{2} = 2}$$

```

int n=5;
for (int i=0; i<n; i++)
{
  for (int j=0; j<n; j++)
  {
    if (i==0 || j==0 || i==(n-1) || j==(n-1))
    {
      System.out.print("*");
    }
    else
    {
      System.out.print(" - ");
    }
  }
  System.out.println();
}
  
```

→ we initialize $n=5$; because 5 rows, 5 columns

$i = \boxed{0}$ — refer to program ① breakdown
 $j = \boxed{0}$

here we are writing conditions based on where we want star '*' and Space '-'

As per program ③ our condition is

$i==0 || j==0 || i==(n-1) || j==(n-1)$ — here we want '*'

remaining we want - Spaces.

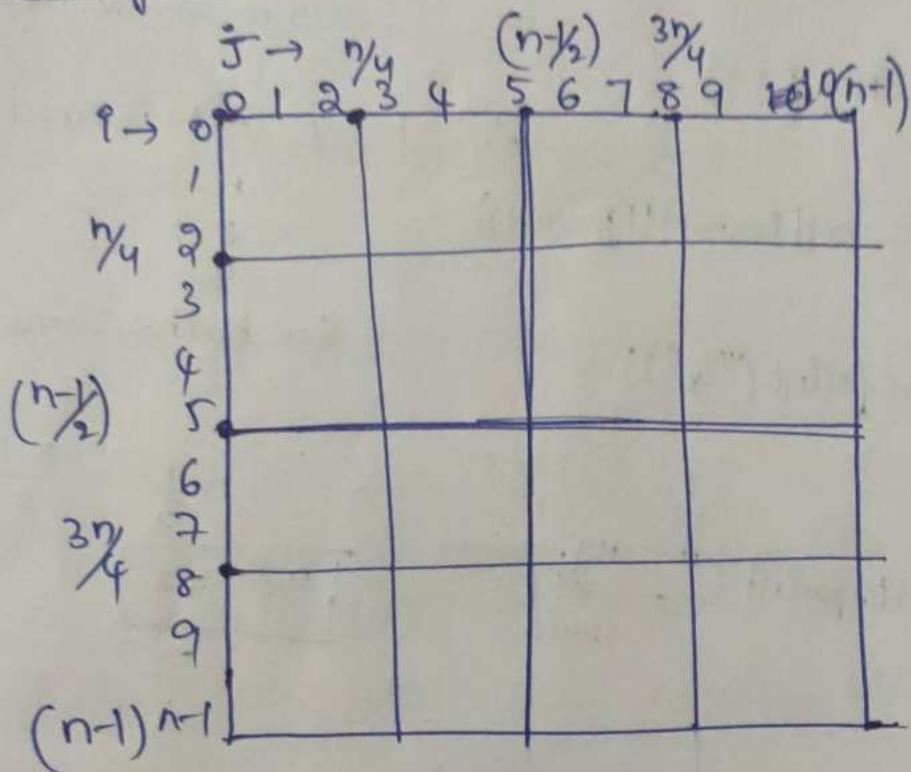
|| - operator — If one true everything true.

&& - operator — If only both true it will execute
 e.g. if one false everything false

Pattern programs

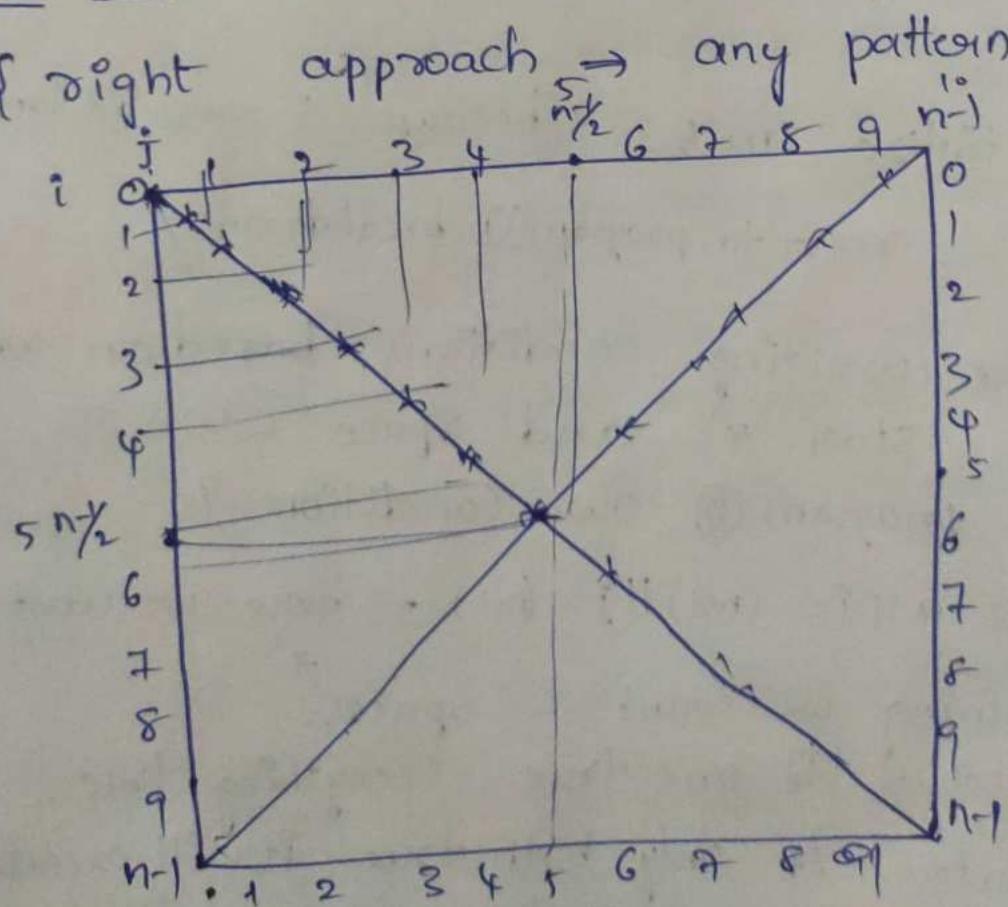
here I am not writing loop overall just conditions
 I will write where we want '*' and "-" spaces
 and stars

Diagram



for Alphabets

→ { digit approach } \rightarrow any pattern }

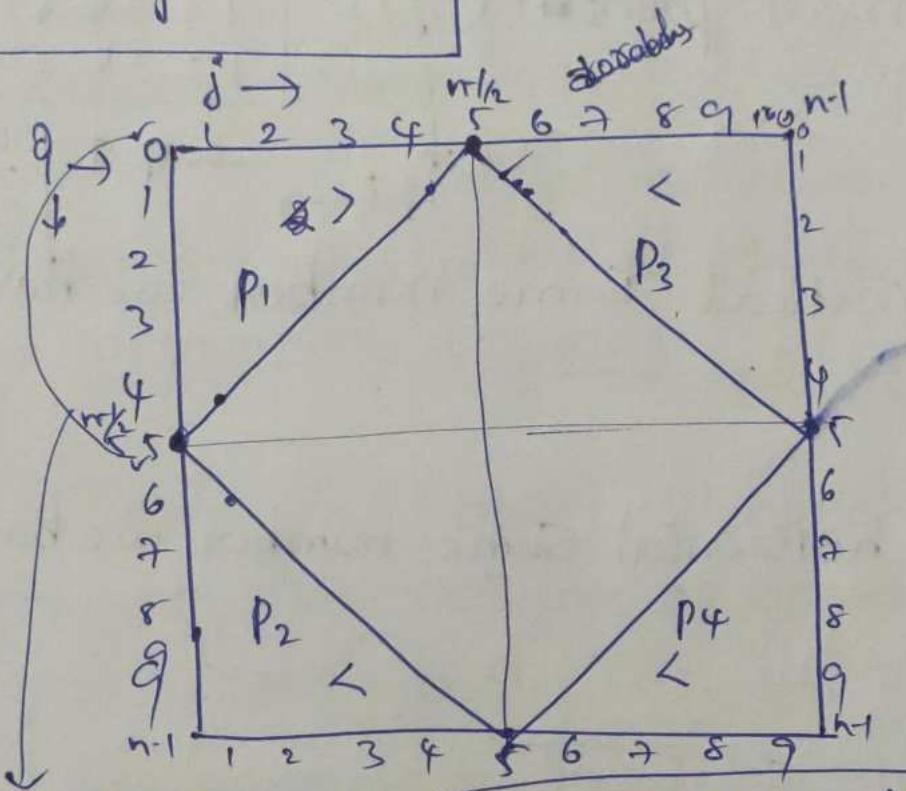


$i = \text{rows}$

$j = \text{columns}$

$$d_1 \Rightarrow i = j$$

$$d_2 \Rightarrow i + j = n - 1$$



$$P_1 \Rightarrow i + j = \frac{(n-1)}{2}$$

$$0+5 = \frac{n-1}{2}$$

$$0+5 = 5$$

$$5 = 5$$

so. print '*'

$$P_2 \Rightarrow i - j = \frac{(n-1)}{2}$$

$$P_3 \Rightarrow j - i = \frac{(n-1)}{2}$$

$$P_4 \Rightarrow i + j = (n-1) + \frac{(n-1)}{2}$$

$$i + j = (n-1 + \frac{(n-1)}{2})$$

Pattern program :-

```
int n=5;
```

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

```
    for (int j=0; j<n; j++) {
```

```
        System.out.print(j);
```

```
}
```

```
}
```

to print vertical same number we have to

print j.

~~to~~ to print horizontal same number we have to
print i.

Output :-

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

Output

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

5 5 5 5 5

OOP's concepts (class and objects)

(Object oriented programming)

By using oop's concept we can solve any
real world - real time problem.

[everything java is in "class"]

Note :-

- * Software means collection of programs
- * programs means set of instructions.
- * To write program set of instructions we need programming language → Java we are learning

① Class → Copy of object/blueprint of object
object → physical existence of any element we say as object Ex: book, car, laptop etc - - -

real time example : Book My show - class
object : person, ticket, chair etc - - -

* In OOPS while solving the problem we need to first mark the objects.

Every Object should have 2 parts.

i) Has- part / fields / attributes (store information as variables)

ii) Does - part / behaviour (represent them as methods.)

Has part → Indicates what it can "hold"
Does part → Indicates what it can "do"

Eg: Student

// Has part (Variables/identifiers)

int id;

String name;

// Does part (Method)

void play() {

 logic ≡

}

void study() {

 logic ≡

}

3) To represent an object, first we need have blue print of an object.

what is blue print and how to represent it.
In Java to represent a blue print we have reserve key word "class".

Eg: class Student{

 int sid;

 String name;

 void play();}

 void study();}

}

Conventions followed by Java developers while writing a class is:

a. class name should be in "pascal convention".

eg: BufferedReader, Student, FileReader

b. Variable name should be in "Camel Case"

e.g.: Name , gender BufferReader
lower lower low only from 2nd word capital.

c. method name should be in "camelCase"

Eg: play, toUpper(), toLower() {lower, 2nd word}

and word ^{upper} should start with Capital letter.

4) We use "new" keyword to create an object
for blue print (class)

class Name Variable = new class Name();

datatype

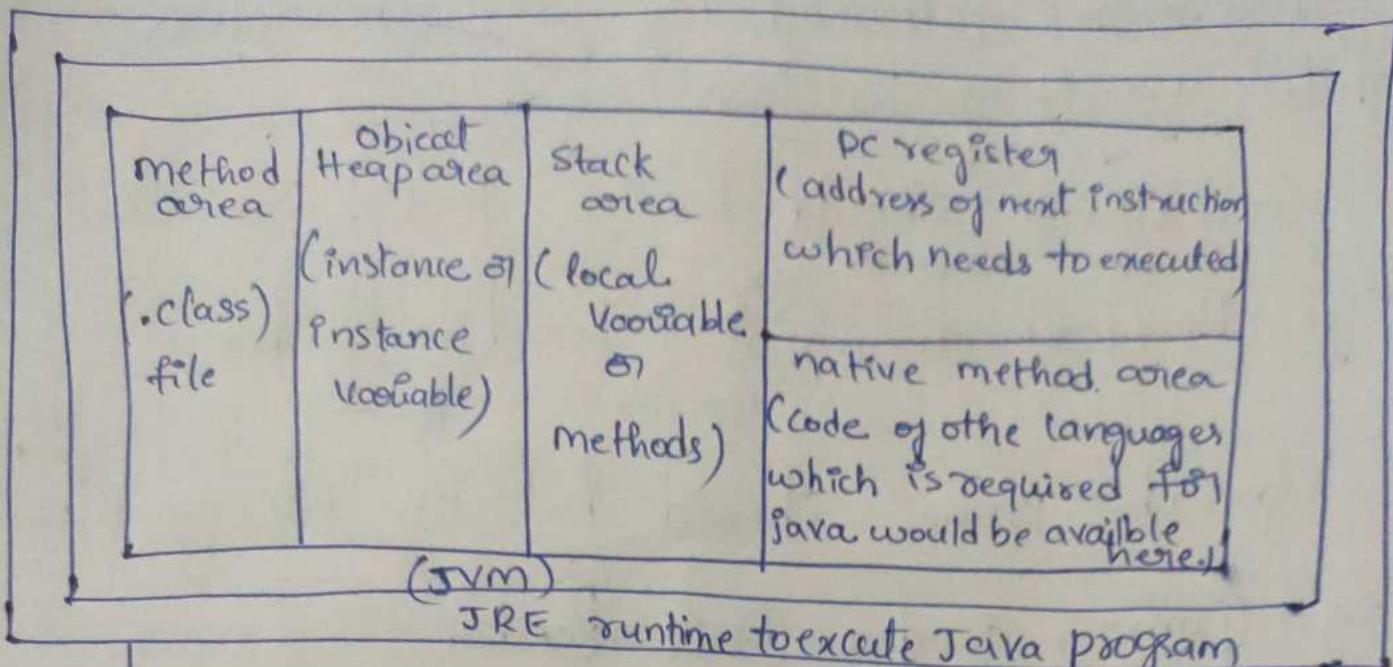
(it is also called Reference

datatype means user
only defining datatype).

reference Variable

Object Creation

↓
Creating object
for className
by using new
keyword.



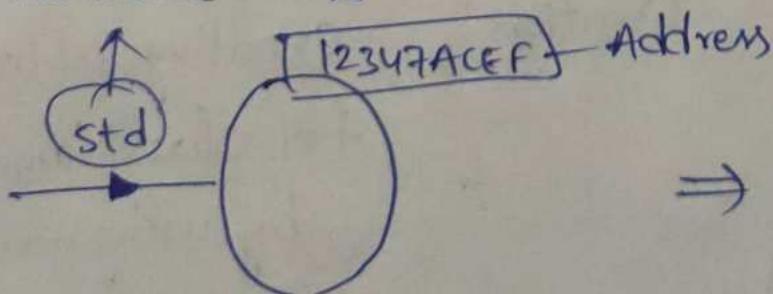
↓ OS allocates this space to execute Java program)

`new` ⇒ it is a signal to (JVM) to create some space for the object in the heap area.

Tell the className, we inform the className, JVM creates object and sends the "hash code" to user

User should collect hash code through "reference variable".

Variable name



⇒ Student std = new
Student();

Object (Student)
(heap area)

5. Every object should always be in constant interaction } Discuss later.
6. Useless object doesn't exist

Types of Variables

int a=10; (primitive datatype)

Student std = new Student(); (reference datatype)

primitive datatype / primitive Variables

primitive Variables can be used to represent primitive Values .
Eg: int x=10; | primitive = predefined by language and it is using some "Keywords" like byte , int etc.

Reference Variables

Reference Variables can be used to refer object
Ex: Student s = new Student();

Instance Variables

If a Variables declare inside class outside method is known as instance Variable
(*)

If the value of the Variables changes from Object to object then, Such Variables are called "Instance Variable".

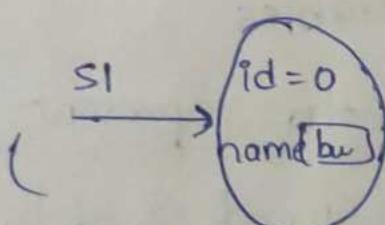
Eg:-

Student S₁ = new Student(); // id=10; name=Sachin

Student S₂ = new Student(); // id = 7; name=dhoni'

To know how many object are there. we can find by using new

1234ABCD → address



To collect
hashcode

Student(heap area)
(JVM)

default Values

int=0

String=null

float=0.0f

boolean→false

Char→space

Note:- Scope of Instance Variable would be available only when we have reference pointing to object. If the object reference becomes null, then can't access "Instance Variables".

Local Variables

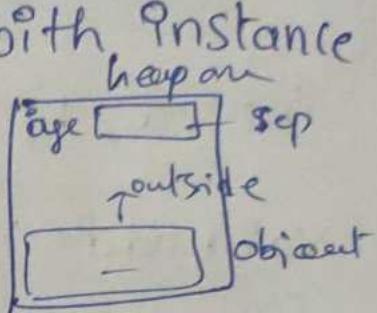
Inside a method if any Variables are declared is known as local variables memory would be allocated in stack area.

Local Variables default will not given by JVM
Values should give by programmer otherwise it
will leads to (CE - compilation error)

Static - Variable.

If we use static key word with instance
Variable

Static int age;



JVM will allocate memory at class loading
time in heap area. without Object creation

~~variables~~

~~Methods }~~: Stack area \leftrightarrow heap area

(Variables/fields)

\rightarrow any task / any work
methods come to picture.

\rightarrow ① name

ex: $\text{return}^{\text{type}} \text{name} (\text{parameters})$

② Input (parameters)

{

③ body

Activity/Body

④ return type.

}

methods we can access 4 types.

⑤ with memory level.

\rightarrow where we are using doesn't matter } Variables
where we declared it is matter.

rule 0 → It should have name, parameters, body, return type

Method 1: Normal

(method any task / to execute that code in

to stack area)

```
class Calculator1  
{
```

```
    int a, b, c;
```

```
    void add()  
    {
```

```
        a=10;
```

```
        b=20;
```

```
        c=a+b;
```

```
        System.out.println(c);
```

```
}
```

```
}
```

```
class LaunchCalc
```

```
{
```

```
    public static void main
```

execute main method one record will create instack area called stack from

```
{
```

```
    Calculator1 calc = new Calculator1();
```

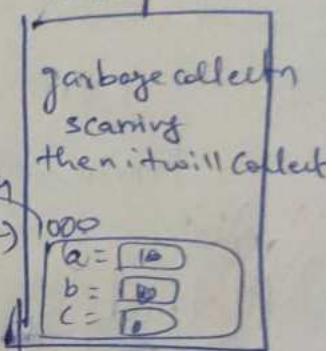
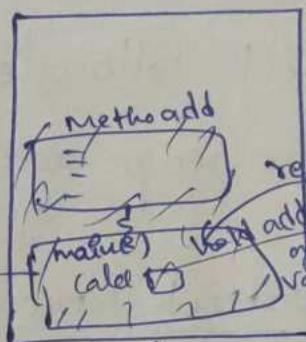
```
    calc.add();
```

control will go to add method.

once method task is over it will delete

then main second delete.

Stack



Stack
frame
of
main
method

JVM

to execute main method one record will create instack area called stack from

→ method 1
→ two important data area are there
stack and heap area

→ After compiling byte code given to JVM

→ JVM to execute the body of main method. (JVM is calling main method)
on stack area

Main method record will be created. That record technically we call it as stack frame

First line in main method object creation is there so. on heap area object will create memory for all instance variables will allocate.

Right part over reference variable object creation word.
Calculator calc = new Calculator();
left part right part

in java = means
 $a=b$ - means
right to left

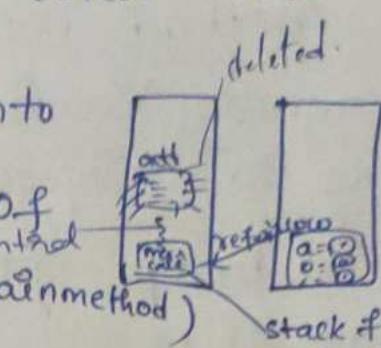
Now left side part reference Variable calc is there memory will allocated in stack frame main method record. ① right to left that address of instance variable referring to reference Variable

Next line add method is there. Then it will go to add method. And again in stack area one more addmethod record will created.

Whatever body it is it will execute.

After task gets over that record will delete.

Control/thread will come back again where it went from. And to main method.



→ if there is no any task in main method.
So main method record also will delete
that reference Variable also delete. then no one
is referring to object.
So, then in heap garbage collector keep on scanning
is there any thing. if it finds any object which
~~no one is~~
~~not~~ referring then it will collect. memory
deallocated. (expecting not retaining)
method 2 → parameters

class Calculate1

{

int res;



void add (int a, int b)

{

res = a + b;

s.o.p (res);

}

}

class launchCalc {

public static void main (String [] args)

{

Calculate1 calc = new Calculate1();

given argument

calc.add (10, 20);

calling fellow

}

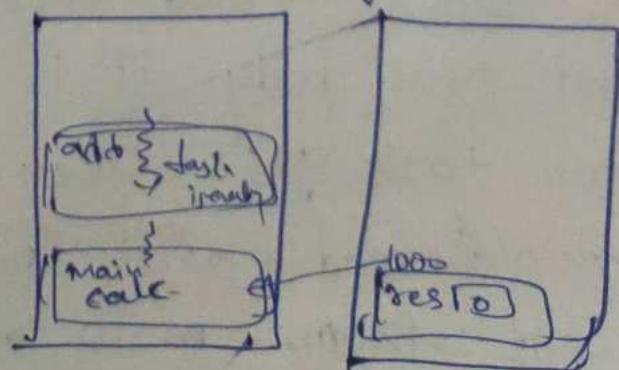
given : argument is passing
taken : parameter is creating

→ in method we are passing
expecting something
So, when someone is calling
method
who is called that fellow
should pass the parameters
which type method is
accepting.
like method 2 - example.

control of execution
and date also

30

memory level



Method 3 returning

→ class Calculator1

{ int res, a, b;

int add() {

 ↑
 return type
 a = 10;

 b = 20;

 res = a + b; return

 }

return res;

→ If this method expecting anything but returning something in int form so when we call method it will return. we can create variable to store or ignore we can use later for our use.

class LaunchCalculator {

 public static void main(String[] args) {

 Calculator1 calc = new Calculator1();

 (Data Type) int c = calc.add();

 System.out.println(c);

}

}

 }

 who is calling method
 it is not expecting any parameter
 so idint write

(giving you something
for who is calling)

it is upto you if you
want take or ignore it

Method 4 :

both passing arguments
returning something also.

class Calculator1 {

 int a, b;

 int add(int a, int b)

{

 int res = a + b;

 return res;

}

```
public class LanchCalc {
```

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

```
        Calculator1 calc = new Calculator1();
```

```
        int res = calc.add (10, 20);
```

```
        System.out.println (res);
```

```
}
```

```
}
```

(garbage collector will deallocate
memory in heap area.)

Method Overloading: (one is to many
polymorphism)

⇒ Methods → stack area and heap area

method overloading

class. Calculator1

```
{
```

```
int add (int a, int b) {
```

```
    return a+b;
```

```
}
```

```
int add (int a, int b, int c) {
```

```
    return a+b+c;
```

```
}
```

```
float add (int a, float b) {
```

```
    return a+b;
```

```
}
```

refers to process of writing
more than one method with same
name and with different
parameters within

Same class.

method overloading

← (false polymorphism)

(one method
one activity)

```
float add (float a, float b) {  
    return a+b;  
}
```

```
float add (int a, float b, float c) {  
    return a+b+c;  
}
```

```
double add (int a, int b, double c) {  
    return a+b+c;  
}
```

```
double add (double a, double b, double c) {  
    return a+b+c;  
}
```

```
}
```

```
public class LaunchMO
```

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

```
    calculator calc = new calculator();
```

```
    int a=10, b=30, c=20
```

```
    float m=10.5f, n=20.5f, o=30.5f
```

```
    double x=15.5, y=25.5, z=35.5;
```

calc.add(^{method overloading})

```
System.out.println (calc.add (a,b));
```

```
System.out.println (calc.add (m,n));
```

```
System.out.println (calc.add (x,y,z));
```

3

The moment calc.add
All the add methods called.
I want expect two inputs.

Early binding

compile time

Polymorphism.

Based on

- 1) no. of parameters
- 2) datatype of parameter
- 3) order of datatype parameter

compile time polymorphism.

(to reduce complexity) (inbuilt methods also)

Compiler resolve the conflict.

a) Number of pa

overloading :-

nothing is overloaded
just illusion.

name and parameter matters.

Return type has no role to play, it's only
method name & parameters.

method overloading with numeric type promotion
Implicit typecasting

Can we overload main method in Java?

Yes, we can overload main method

however JVM will call such a main method

which accept (String[] args) - JVM will take this
Search for this.

~~28/10/22~~ Array → why what How Arrays treated as objects.
to create use "new" keyword.

→ Variable → to store data int/

I want to store 5 integer Values:

int a;
int b;
int c;
int d;
int e;

 } "each variable store only value"
 } → 5 variables that too different
 } variable names.
 } "So what if we want to
 } store 20, 100 values with
 } different variable names." not good

its difficult to create or write. its a length process.

→ convenient / traditional way to store data is to create Variable.

→ So Array came into picture.

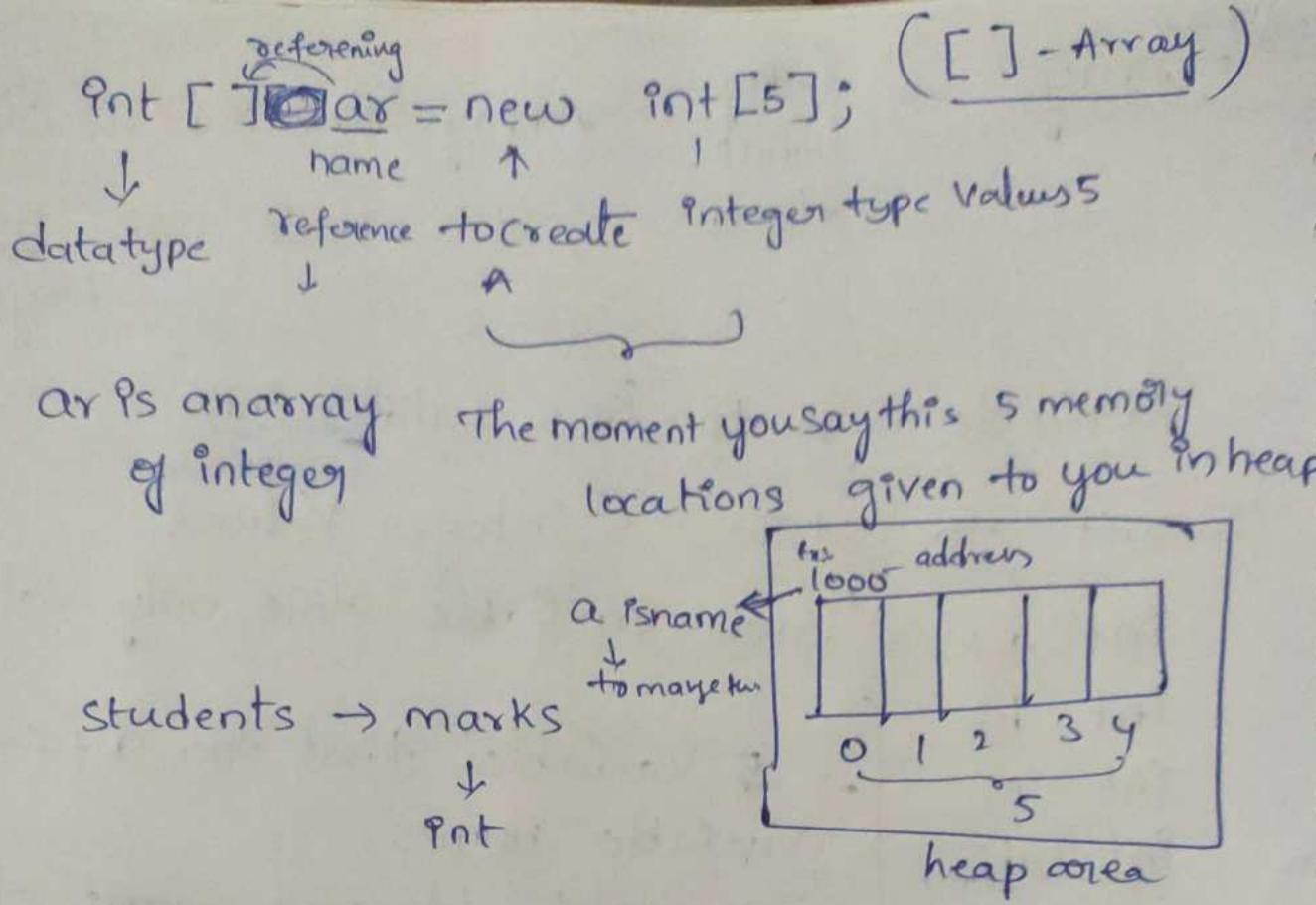
↓
Collection
of
data

→ "Array" is indexed based data structure to store large volume of data using single variable name.

In Java Array store only homogenous data

→ Array is in Heap Area,

because Java treats Array is Object.

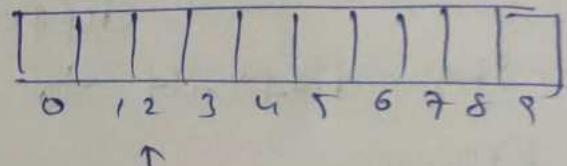


$\text{int [] ar} = \text{new } \text{int}[10];$

ar is array of int

$\text{ar}[2] = 40;$

$\text{ar}[8] = 70;$



console

10

`System.out.println(ar[2]);`

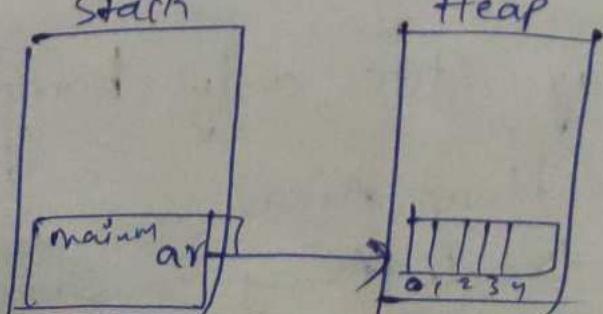
(Arrays are two types 1) regular 2) Jagged)

Case 1: To store marks of 5 Students

\Rightarrow Students
 5
 int type
 Stack

$\Rightarrow \text{int [] ar} = \text{new } \text{int}[5]$

ar is reference array of integers



to create array of int type 5 val

Case 2 To store marks of 3 classes each with 4 students
 2D arrays [2D regular array]

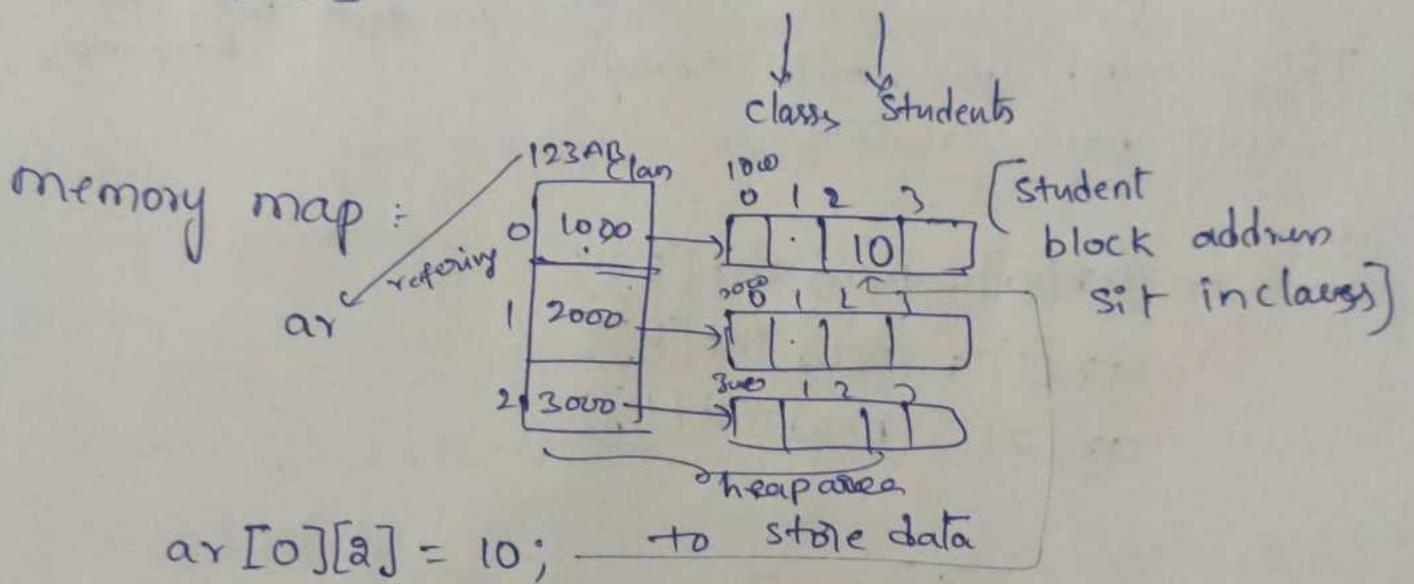
In this I have classes students

Regular -

0	→	4	regular data Same data in every class
1	→	4	
2	→	4	

2D - Array creation

`int [][] ar = new int[3][4];`

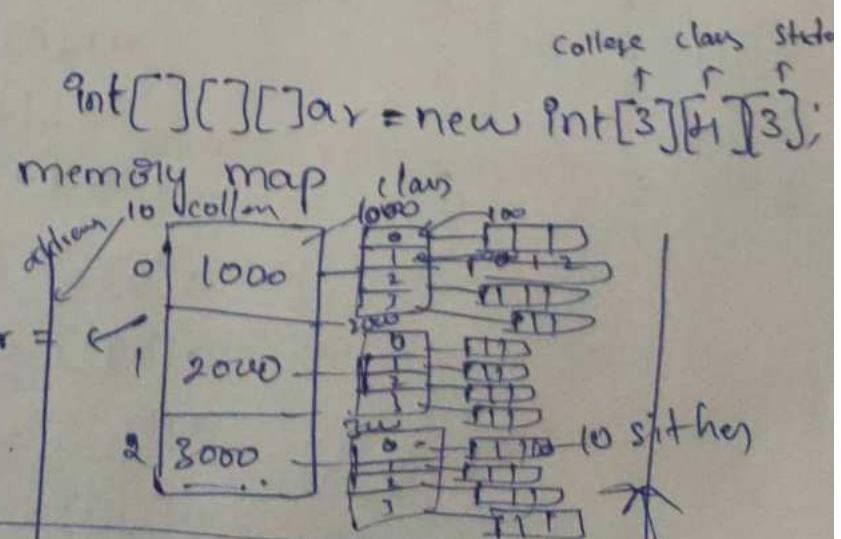


case 3 3 college 4 classes 3 students

3D - 3dimension array regular same data

College	Classes	Students
0	0 1 2 3	3
1	0 1 2 3	3
2	0 1 2 3	3

`ar[2][0][2] = 10;`



Case 4

Classes	Student	(2D joggded Array)
0	5	
1	3	joggded - random
2	4	not similar.

to create

`int[][] ar = new int[3][empty]`

↓

we don't have similar
keep empty.

This much you write Array
will not create

So.

`ar[0] = new int[5];`

`ar[1] = new int[3];`

`ar[2] = new int[4];`

`ar[0][4] = 10;`

3D - joggded Array

College	Class	Students.
0	1 2	4 3
1	0 1 2 3 4	?
2	0 1 2 3	3 4

`int[][][] ar = new int[3][2][4];`
its not sufficient

`ar[0] = new int[2][4];`

`ar[1] = new int[4][3];`

`ar[2] = new int[3][2];`

`ar[0][0] = new int[4];`

`ar[0][1] = new int[2];`

`ar[1][0] = new int[3];`

`ar[1][1] = new int[1];`

`ar[1][2] = new int[5];`

} using
loop.

→ Array introduction ✓

⇒ User input from console :-

class Scanner { → import java.util.* package
{

 nextInt(); } } - many methods are there.
 nextLine(); }

}

Scanner scan = new Scanner (System.in);
 ↑
 ↓ something into program
int a = scan.nextInt();

store

1D Array regular

(arr.length to find array length)

import java.util.*;

class Launch1

{

public static void main (String [] args)

{

 int [] ar = new int [5];

 Scanner scan = new Scanner (System.in);

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

 System.out.println ("please Enter marks of student" + i);
 ar[i] = scan.nextInt();

 System.out.println ("The marks of student");

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

 System.out.print (ar[i] + " ");

}

QD Array (Regular) by using for loop

class student

0 4
1 4
2 4
 ↑
 ID

↓
 j

Scanner s = new Scanner (System.in);

int cls = s.nextInt();

int std = s.nextInt();

int [][] a = new int [cls][std]

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

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

 s.o.p (" Enter class "+i+" "+ "student "+j);

 a[i][j] = s.nextInt();

}

}

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

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

 s.o.p (a[i][j] + " ");

}

}

—o—

Same for jagged array also but
we will not define student while creating
array.

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

a[0] = new int [3];

a[1] = new int [2];

remaining same process

3D Jagged Array.

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

```
a[0] = new int [2][ ];
```

```
a[1] = new int [3][ ];
```

```
a[0][0] = new int [4];
```

clap cls
0 0 - 2

random.

std
4,3

1 0 - 3
2

2,3,4

```
a[0][1] = new int [3];
```

```
a[1][0] = new int [2];
```

```
a[1][1] = new int [3];
```

```
a[1][2] = new int [4];
```

How Array Secure in Java?

RAM is a Collection of Objects bytes.

Buffer OverRun

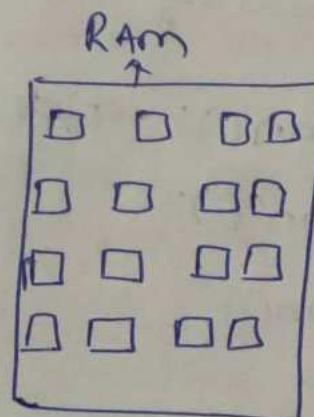
```
int [3]
```

```
a[0] = 1;
```

```
a[1] = 20;
```

```
a[2] = 30;
```

```
a[3] = 40;
```



Here 3 locations are given if we access for a[3] the values will be overwritten is called BufferOverRun ex:- c,

In Java no BufferOverRun.

Because array guarded with a boundaries.

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

```
a [0] = 10;
```

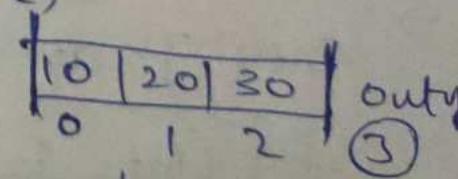
```
a [1] = 20
```

```
a [2] = 30
```

```
a [3] = 40; =>
```

Array Index out of boundary -

Runtime error / Exception



Creating array for object

```
class Fan {
    int cost;
    String brand;
    int noofWings;
}
```

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

Fan [] f = new Fan[3]

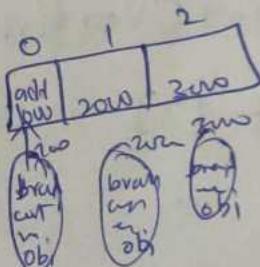
|
reference

How we are creating int type like fan type

f[0] = new Fan(); - object

f[1] = new Fan();

f[2] = new Fan();



f[0].brand = "Vijay";

f[1].cost = 10000;

Reverse

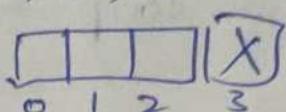
→ int [] ar = { 10, 20, 30, 40 };

for (int i = ar.length - 1; i >= 0; i--)

{
 s.o.p (ar[i] + " ");

}

Disadvantages of array

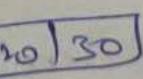
- It can store only homogenous type of data
 - Memory of array is fixed in size
(it cannot grow or shrink)
- int [] a = new int[3]; 
existing array will not increase.
another location will create
- Array demands contiguous memory locations.
- int a[] = new int[3];
4 bytes × 3
= 12 bytes

Array will not make use of dispersed location
it was contiguous memory location
it will count 0, 1, 2, 3 like if in 4th place some
value is there then it will again count 0, 1, 2 like
after 4th location. So, this is disadvantage.

Another way of creating array

int [] a = { 10, 20, 30 }; - we create like this

for each loop

a → 

for (int ~~variable~~ ^{name}; array name) { giving / it will talk without start and end no need.

s.o.println(* variable);

automatically it will take.

}

10 20 30 - output

→ Objects → Heap

arrayname

arr	→	10	20	30	40	50
		0	1	2	3	4

Strong core + JEE ⇒ Spring }

→ Enhanced for loop → for each }

Position to position
we can create
loop

for (datatype Variable : Array name)

{

- S.O.P (variable)

}

Traversing / iteration while travelling getting values.

iteration - variable - body

2D - iteration

- ① we don't access index
- ② it will traverse / iterate
- only in forward direction.

int [][] a = {{10, 20}, {30, 40, 50}, {60, 70, 80, 90}};

for (int elem : a) { *entire row is getting*

for (int ar[] : a) { *storing here*

for (int elem : ar) {

 S.O.P (elem);

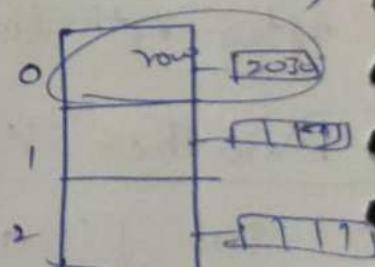
}

iteration on row.

}

first one row do iteration

and 2nd row and next 3rd row



int [] x; ✓
 x = new int[4]; ✓ int [6] ar; ✗ not valid

int [] x; ✓
 int x[]; ✓

int [] ar[];
 int [][] a;
 int [] [] b;
 int [][] c;

int [] x[], y[];
 x is two dimension
 y is one dimension

int [][] xy, yz;
two are dimension

int [] ab, cd;
 ab = new int[4]; ~ Valid
 cd = new int[5];

int m, n, o;
 m = 10;
 n = 20; ✓ Valid.
 o = 40;

int [][] aa, bb[];
 32
20

after ';' not } java not allowed

int [] a = new int[5] size ✗ long, float, double
 ↓
 (accr runtime) ✓ by, short, int, char

↑
 Expection in Array Compiler Syntax.

class Demo { }

}

if class is not there can't create object.

Every type of array will have specific class
 inbuilt classes

Demo d = new Demo();

int [] a = new int[];

That class is not for programmer }

→ Arrays - So many methods are there.

↳ sort

→ fill

↳ binary search

}

utility class - Arrays.

Static methods - without creation of object I can call method.

Arrays.fill(a, 5);

Sorting } Arrays.sort(a);

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

int sum = 0;

temp

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

sum = sum + a[i];

Sum += a[i];

int [] ar = {25, 30, 40, 70, 10};

int max = ar[0];

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

 if (ar[i] > max)

 {

 max = ar[i];

 }

 System.out.println(max);

→

1ST NOV

linear search is used to search a key element from multiple elements. Linear Search is less used today because it is slower than binary search and hashing.

Searching

Linear Search



Sequential Search

Binary Search

- Step 1 - Traverse the array
2 - Match the key element with array
3 - if key found return index position
4 - if key element not found
 return -1

whether element is present or not in collection

using a linear search algorithm

if you want travel

→ traversing / travelling

public class Linear Search {

 public static void main (String [] args)

{

 int [] ar = {10, 20, 30, 40, 50, 60, 70, 80, 90};

 boolean flag = false;

 Scanner scan = new Scanner (System.in);

 System.out.println ("Enter the key to be searched");

 int key = scan.nextInt();

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

 { if (key == ar[i])

{

 System.out.println ("key" + key + "found at index" + i);
 flag = true;

 break;

}

; if (flag == false)

{ System.out.println ("Key not found");

};

Binary Search — Array should be sorted.
1st Step — Array must be sorted.

public class BBS {

```
public static void main(String[] args){  
    int[] ar = {10, 20, 30, 40, 50, 60, 70, 80, 90};  
    Scanner scan = new Scanner(System.in);  
    int key = scan.nextInt();  
    int low = 0;  
    int high = ar.length - 1;  
    while (low <= high) {  
        int mid =  $\frac{low+high}{2}$ ;  
        if (key == ar[mid]) {  
            System.out.println("key " + key + " found at index  
" + mid);  
            break;  
        }  
        else if (key < ar[mid]) {  
            high = mid - 1;  
        }  
        else if (key > ar[mid]) {  
            low = mid + 1;  
        }  
    }  
    if (low > high) {  
        System.out.println("Key not found");  
    }  
}
```

Bubble Sort

Public class LaunchBSQ

Public static void main (String[] args){

int [] a = {7, 5, 2, 3, 1, 4, 6};

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

{

for (int j=1; j<a.length-i; j++)

{

if (a[j] < a[j-1])

{

int tempVar = a[j]

a[j] = a[j-1]

a[j-1] = tempVar;

}

}

}

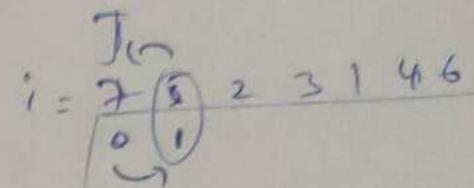
for (int el: a)

{ System.out.print (el + " "); }

}

}

}



logic for bubble sort

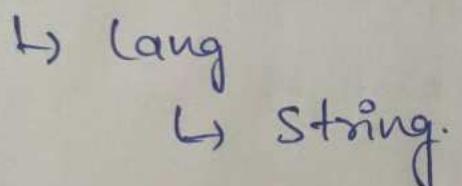
Strings in Java

- A String Variable Contains a collection of characters surrounded by double quotes.
- it is classname , String is basically an inbuilt class for which object can be created (java.lang).

Stole of String (Java.lang.String)

[package].

Java



String = Collection of characters

String = "Vijay"; — ① type

String s₁ = "Vijay"; ②

String s₁ = new String ("Vijay"); ③ type

Java.lang.String

→ String it refers to an Object in Java present in package called java.lang.String(c)

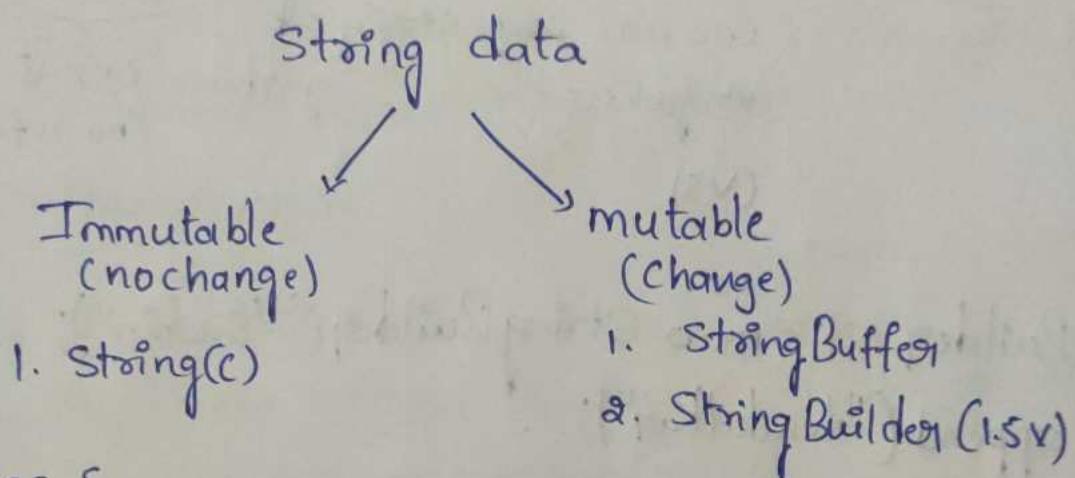
→ String refers to collection of characters.

eg:: String s= "Sachin";
 System.out.println(s); // Sachin

String s= new String ("Sachin");

System.out.println(s); // Sachin.

In Java String Object is by default immutable, meaning once the object is created we cannot change the value of the object, if we try to change then those changes will be reflected on the new object not on the existing object.



```
class {  
    // instances Variables  
    // methods.  
}
```

Case 1 :

```
String s = "Sachin";  
s.concat("tendulkar"); // (new Object got  
                        created with modification  
                        so immutable).
```

Output : Sachin
(Vs.)

```
StringBuilder sb = new StringBuilder ("Sachin");
```

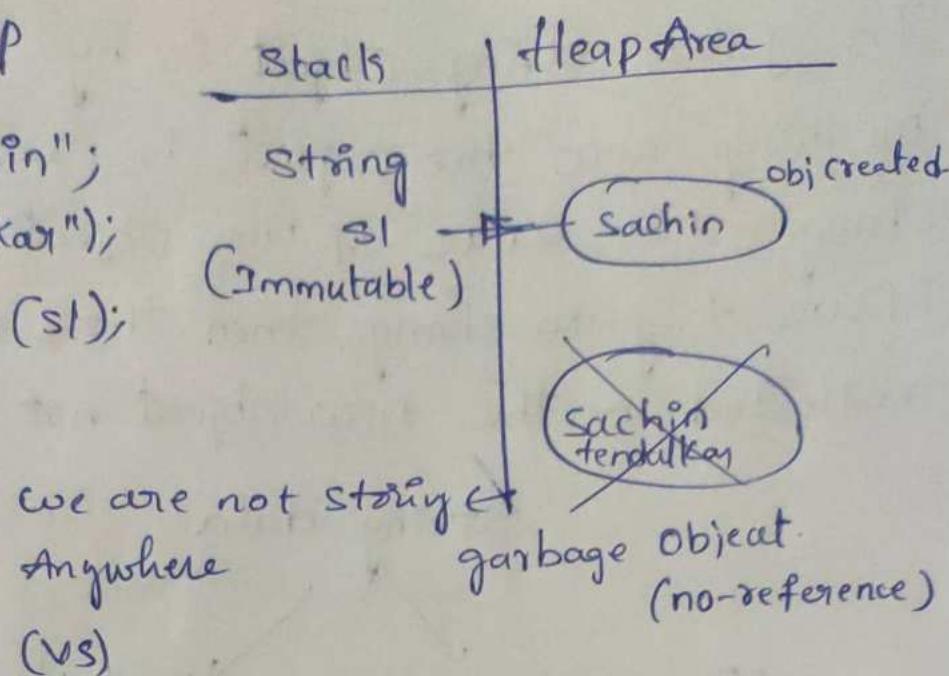
```
sb.append ("tendulkar"); // (on the same object
```

```
System.out.println (sb);           modification so mutable)
```

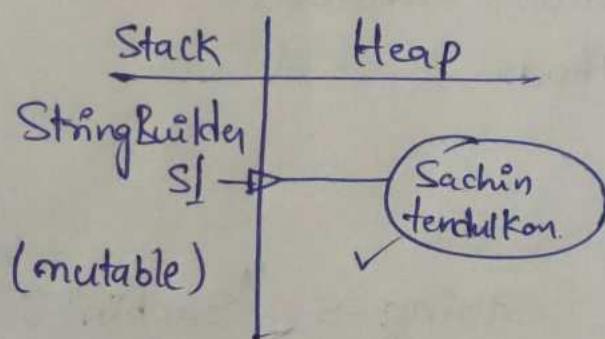
Output : Sachintendulkar,

Case 1 memory map

```
String s1 = "Sachin";
s1.concat("tendulkar");
System.out.println(s1);
//Sachin
```



```
StringBuilder s1 = new StringBuilder("Sachin");
s1.append("tendulkar");
System.out.println(s1); //Sachintendulkar.
```



Case 2:

```
String s1 = new String("Sachin");
String s2 = new String("Sachin");
System.out.println(s1==s2); //false
System.out.println(s1.equals(s2)); //true
⇒ String Class.equals method will compare the
content of the object if same return true
otherwise return false
(v/s)
```

StringBuilder sb1 = new StringBuilder ("Sachin");

StringBuilder sb2 = new StringBuilder ("sachin");

System.out.println (sb1==sb2); // false.

System.out.println (sb1.equals(sb2)); // false

→ StringBuilder class . equals method for reference comparison if different object returns false, even if different object returns false ..

Case 2:- Memory map

String s1 = "Sachin";

String s2 = new String ("Sachin");

System.out.println(s1 == s2); // false ^{↑ compare references only}

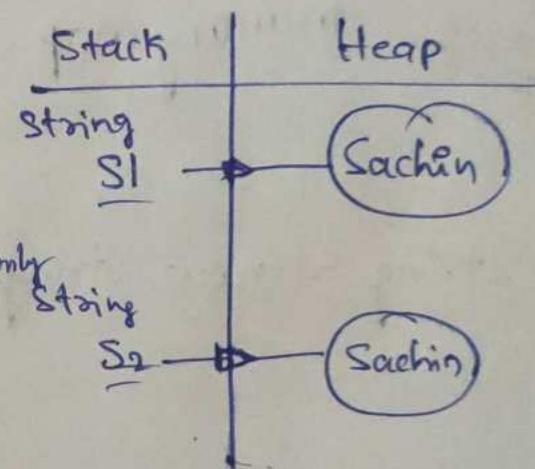
System.out.println (s1.equals(s2));



String class equals ()

will compare the content present inside the string.
"Sachin".equals ("sachin")

true



StringBuilder s1 = new StringBuilder ("Sachin");

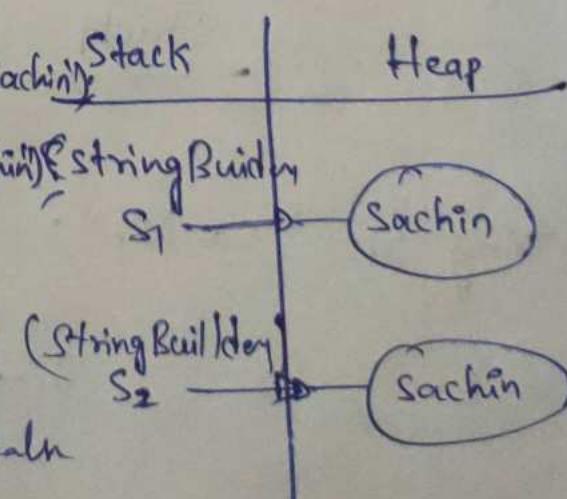
StringBuilder s2 = new StringBuilder ("Sachin");

System.out.println (s1==s2); false

System.out.println (s1.equals(s2));



StringBuilder class equals () compare the reference (address of object) not the content of StringBuilder.
false.



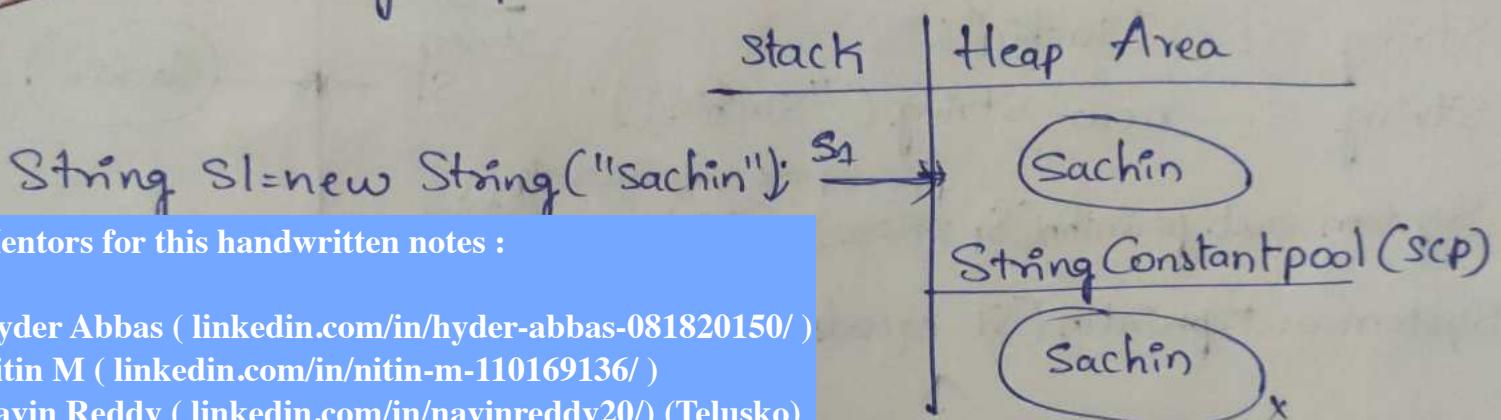
Case 3 :- String s = new String ("Sachin");

In this case 2 objects will be created one in the heap and the other one in the string constant pool, the reference will always point to heap (vs)

String s = "Sachin";

In the case only one object will be created in the SCP and it will be referred by our reference.

case3 memory map



Mentors for this handwritten notes :

Hyder Abbas (linkedin.com/in/hyder-abbas-081820150/)

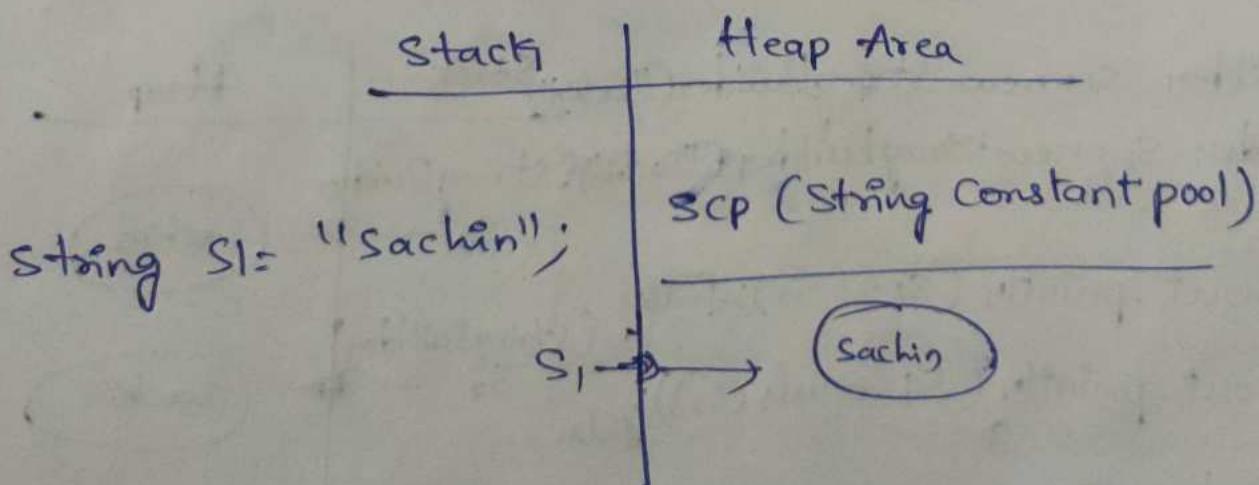
Nitin M (linkedin.com/in/nitin-m-110169136/)

Navin Reddy (linkedin.com/in/navinreddy20/) (Telusko)

written by pallada vijaykumar

@ineuron.ai

Even though it is a garbage Object Garbage Collector can't clean the data present in 'SCP'



Note : Object creation in SCP is always optional, 1st JVM will check is any object already created with required content or not if it is already there then it will reuse the existing object instead of creating the new object.

If it is not available only then new object will be created, so we say in SCP no chance of existing objects with same content. In SCP duplicates will not be allowed.

Garbage collector cannot access SCP area, Even though object does not have any reference still object is not eligible for garbage collector. All SCP objects will be destroyed only at the time of JVM shutdowns.

Eg: `String s1 = new String ("dhoni");`
`String s2 = new String ("dhoni");`

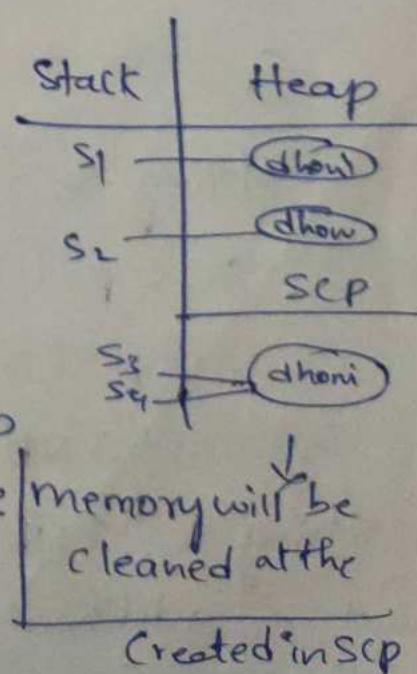
`String s3 = ("dhoni");`

`String s4 = ("dhoni");`

`s1.o.p (s1 == s2); // false`

`s1.o.p (s3 == s4); // True`

Two objects are created in the heap with data as "dhoni" with reference as s₁, s₂. One object is with reference as s₃, s₄.



Case 4

```

String s = new String ("Sachin");
s.concat("tendulkar");
s = s.concat ("IND");
s = "Sachintendulkar";
s.o.p(s);
    
```

Output: Sachintendulkar.

Concat method will Create object in SCP

A method call will be by JVM.

Because of runtime operation if new Object is created it will create in Heap area not in SCP.

Direct literals are created in SCP, Because of runtime operation if Object is required to create compulsary that Object should be placed on Heap but not in SCP.

→ String s₁ = new String ("Sachin");

s₁.concat ("tendulkar");

s₁ = s₁ + "IND";

String s₂ = s₁.concat ("MI");

s.o.p (s₁);

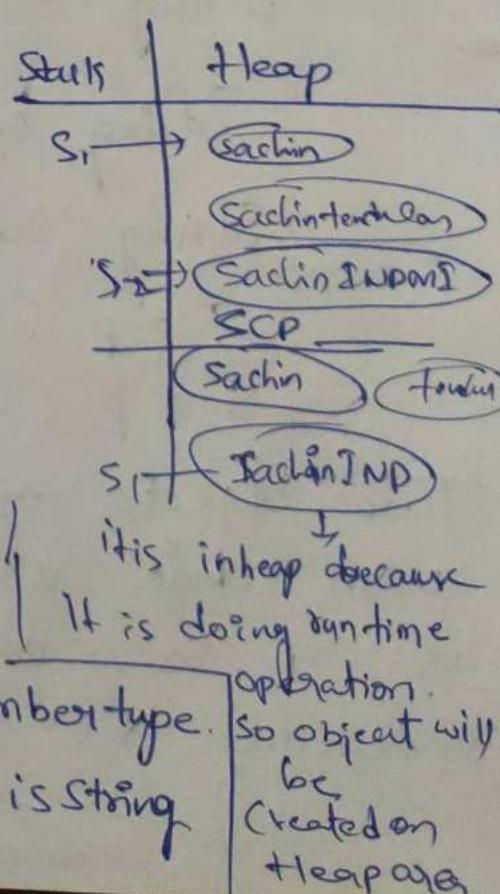
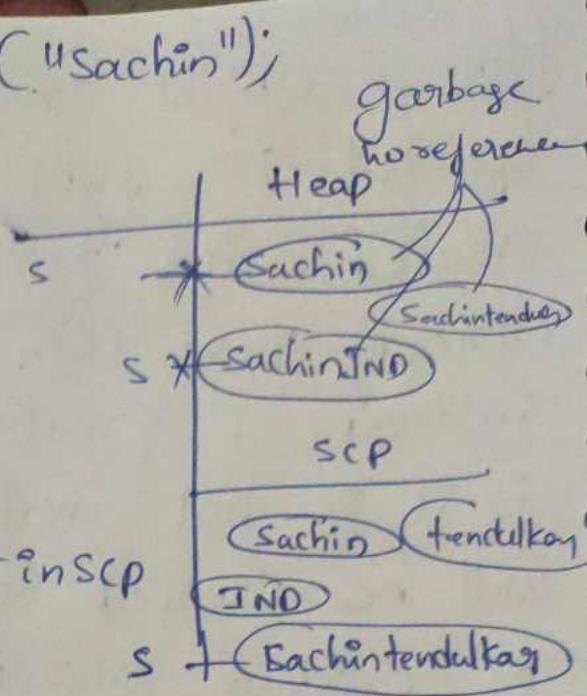
s.o.p (s₂);

s₁ = s₁ + "IND"

String + "IND"

+ - addition both operands are number type.

(+) - Concatination if one operand is String



```
String s1 = new String ("Sachin");
```

```
String s2 = s1. intern();
```

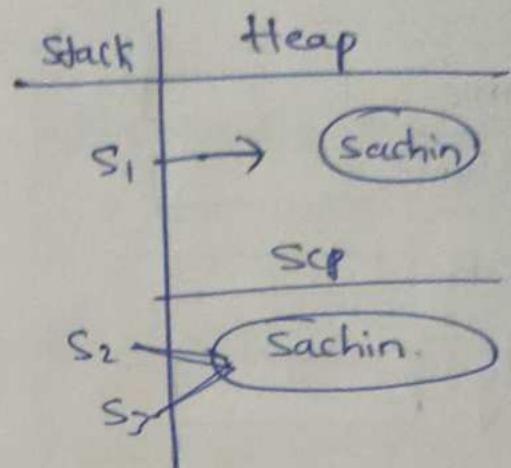
```
s.o.p (s1 == s2);
```

false

```
String s3 = " Sachin ";
```

```
s.o.p (s2 == s3);
```

true



Interning → Using Heap Object reference, if we want to get Corresponding Scp object then we need to use `intern()` method.

```
String s1 = new String ("Sachin");
```

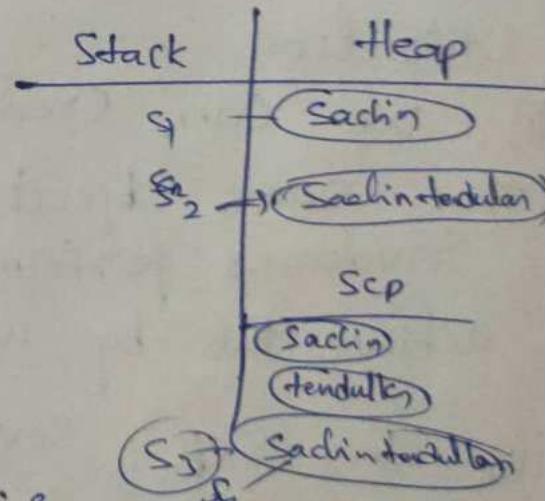
```
String s2 = s1.concat ("tendulkar");
```

```
String s3 = s2.intern();
```

```
String s4 = "Sachin tendulkar"
```

```
(s3 == s4)
```

true



Using heap object reference if we can't get corresponding Scp object and if object does not exists then `intern()` will create new object in SCP and it returns.

importance of Scp

Adhaar Card Application

Name

Immutable
Scp Area

F. Name

Person

DOB

P₁

City

P₂

photo

P₃

P₄

if p₄ changes location Hyd

Heap
mutable
Hyd

i) In our program if any String object is required to use repeatedly then it is not recommended to create multiple object with same content it reduces performance of the system and effects memory utilization.

ii) we can create only one copy and we reuse the same object for every requirement. This approach involves performance and memory utilization we can achieve this by using "Scp".

iii) In Scp several reference pointer to same object the main disadvantage in this approach is by using one reference if we are performing any change the remaining references will be impacted. To overcome this problem Sun microsystem implemented immutable concept for String object.

1) According to this once we create an Object we can't perform any change in existing object if we trying to perform any change a new String object will be created hence immutable is main disadvantage of sep.

API level (Application programming Interface);
Some one wrote the code how he will give .class file end users will use and take the benefit.
Entire java we are learning as API only
if some class name & same method name is known as constructor.

Methods of a string (Important)

1. public char charAt(int index)
2. public String concat(String str)
3. " boolean equals(Object o)
4. " equalsIgnoreCase(String s)
5. " String substring(int begin)
6. " " (int begin, int end)
7. " int length();
8. " String replace(char old; char new)
9. public String toLowerCase()
10. " " " uppercase()
11. " String trim()
12. " int indexOf(char ch)
13. " int lastIndexOf(char ch).