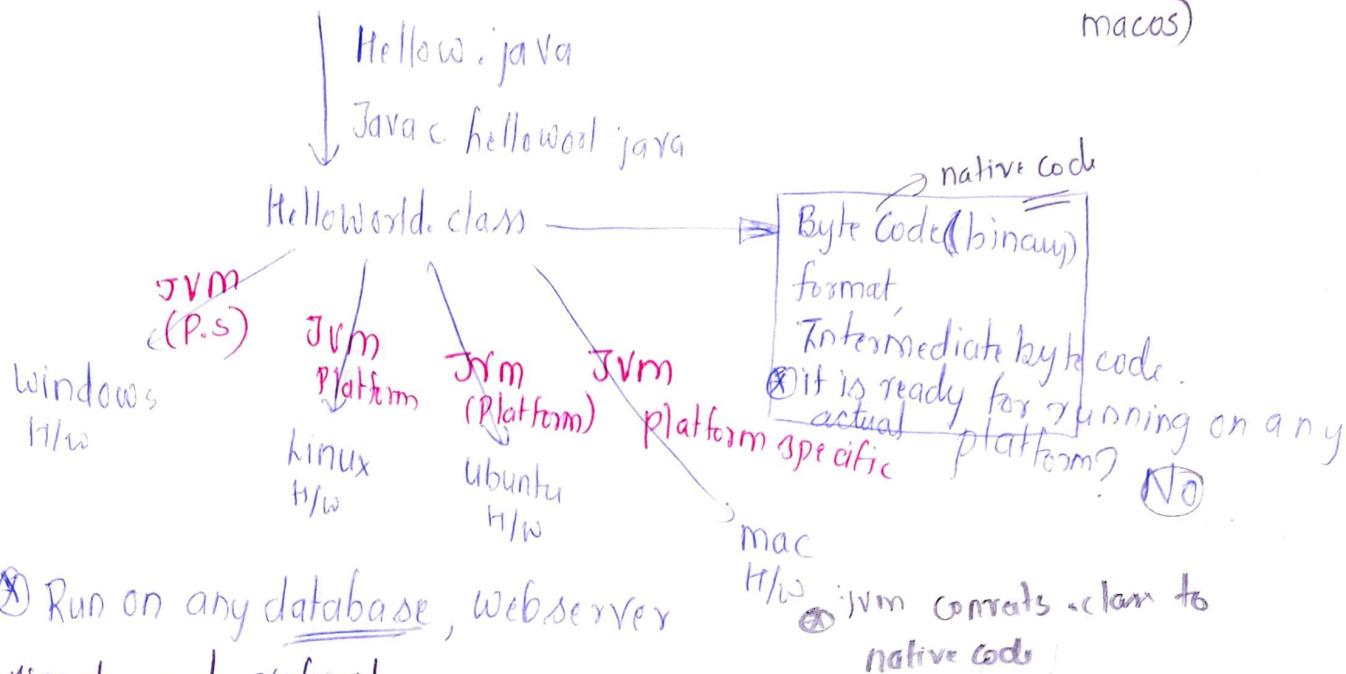


16/10/2021JavaWhy Java

- ① platform independance -



- ② Run on any database, webserver

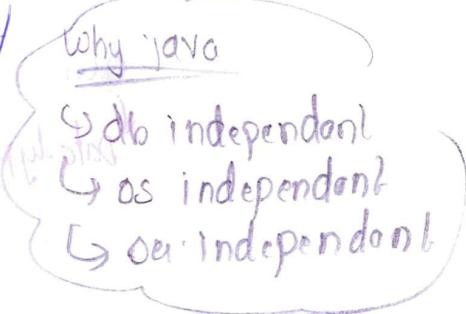
- ③ Simple and robust

- ④ Security ⑤ automatic memory management

Day 1 → SRC
bin

- ⑥ legal access specifier for class?

- ① default (scope access in same folder)
- ② public (scope within same package)

Naming Convention

- ① class, interface, enums → 1st letter of 1st word must be cap and follow camel casing

HelloWorld

- ② Data members / methods → 1st must start with lower case & follow camel case

HelloWorld

- ③ constants = all uppr

199 e.g. class level only → default & public
class HelloWorld { class level
public static void main(String[] args) {
anywhere System.out.println("Welcome 2 Java");
} System.out.println("Hello, " + args[0] + " " + args[1]);
}

JVM → HelloWorld.main();

JDK →

(tools)

• lang package is by default called

package - java.lang

class → System

System.out.

↳ err

in

out → public static final PrintStream out

↳ Data type - printstream.

↳ print

↳ println

Goto folder → cmd.

Commnd → dir → directory content

cls → clear screen.

→ Java c

-d <directory> → Where to generate .class
press tab to autofill file

cd → change director

parent dir ↗ \bin

error → after `javac helloworld.java`

in thread "main" corruspond ArrayIndexOutOfBoundsException
 for length → no member in args
 at HelloWorld.main (HelloWorld.java)

How to pass throw cmd

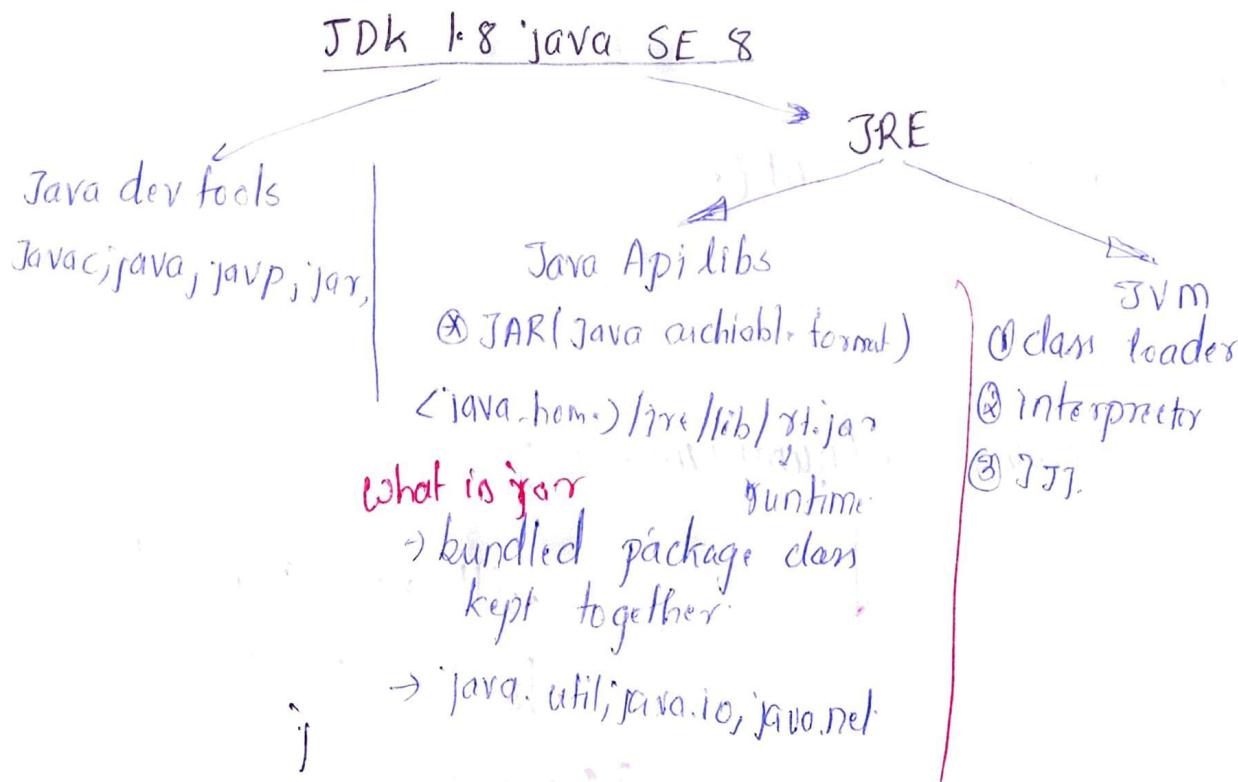
bin>java HelloWorld Madhura, Antukar
 args[0] args[1]

To chang directory

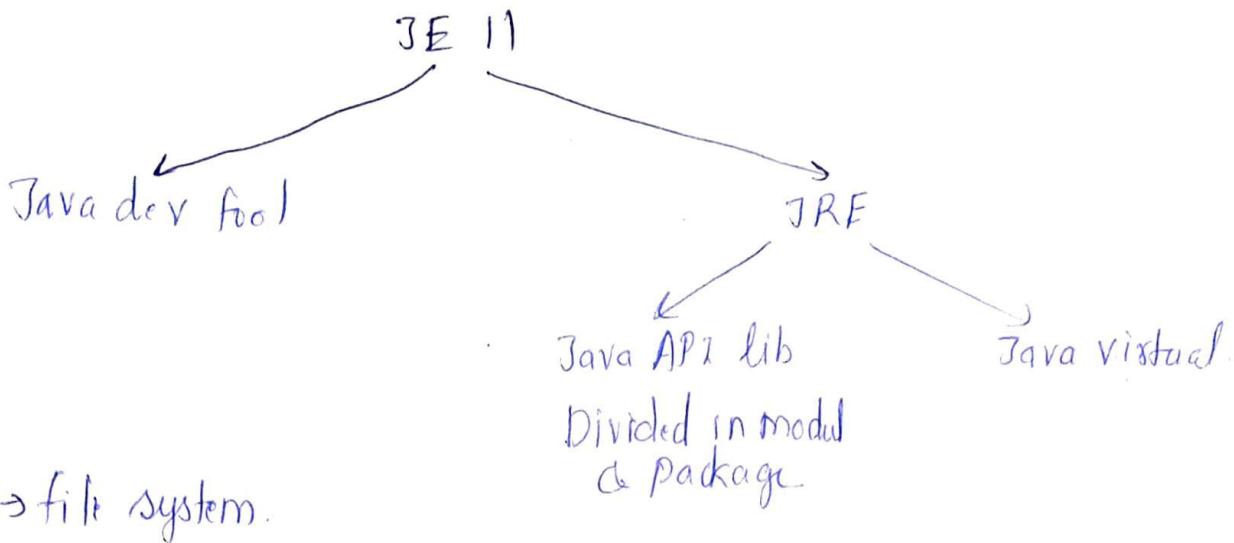
cd .. \src
 ↗ parent folder.

To compil.

javac -d ..\bin HelloWorld.java.
 ↗ parent folder.



20/



Java.sum

* → contains for string.

(" " + args[0] + args[1]) → this string return by
("Result: " + (args[0] + args[1])) → args
Concat "Result: " + "10" + "15" → o/p 1015

Integer Class

public static int parseInt (String s)

Integer.parseInt (string);

⑧ int num1 = Integer.parseInt (args[0]);

int num2 = Integer.parseInt (args[1]);

System.out.println ("Result: " + (num1+num2));

O/P 25

I/P = 10, A → will not convert
to unicode
gives error

I/P = 10 12.35 → will give error

Can we create more than 1 java file in ~~one~~ ~~file~~ 202

class A

{

}

class B

{

3 class public

{

}

{

Interview

Quesⁿ

How to compile.

Javac test.java

generate 3 classes

⊗ one class file for each class declaratⁿ

⊗

Madhura Antuskar

can we write a class without main.

we can compile but we cannot run

⊗ what will happen if we write one main in A
it will compile and as well as run

System.out.println();

public static void main(String[] args);

⊗ what happens if we have multiple mains

it will run but which we call e.g. java A - in A's main
java B - in B's main

Bad/Horrible programming practice to have many mains

Access specifiers of class A, B, C → default/package

⊗ when we have

public class B { for public class

{

= file name should
match

if default = not required.

28' for Default

file name & class name
can be different

for Public
should be same

④ Java Compiler does not allow uninitialised variable

Interview Given code with uninitialised variable:

Monday -> datatype,

18/10/2021

Day 2 names of packages - java.lang

what does package consist: functionally similar classes.

JVM: what are parts of JVM?

① classloader subsystem

② Execution Eng

③ increasing scope order: private → default → protected → public

④ what will happen?

(Scanner sc;

int data = sc.nextInt();

boolean active;

System.out.println("active");

united variable reference
(not even to null):
Compile error

⑤ local to main, hence no link
united uninitialised.

⑥ compile time error
it is not initialised "active")

Given Default means ⇒ no restriction on name of
file and source file.

dotnet

.background = Third();

Data types

204

① Primitive (Built-in type) they hold values.

non-numeric a) Boolean \rightarrow boolean active = false

b) Numeric type

c)

it stores value of active
and not the address

⊗ class type of reference?

interfaces

array type

⊗ all key words with small case

⊗ value Boolean Variable hold?

Ans:- 'True' or 'false'

e.g boolean isSleeping = $\begin{matrix} \times \\ \checkmark \end{matrix}$, true, false

How many bytes allocated to Boolean?

It is as per JVM specific.

1bit, 1byte, 4byte

⊗ Numeric type

① Integral type

a) character type \rightarrow char

b) integer

How many byte of char?

2 byte, it represent unicode 0000 - ffff

1 byte \rightarrow ascii. \hookrightarrow include devnagari script

char is signed or unsigned

unsigned

a) byte, short, int, long

byte for byte, short, int, long

1 byte / 2 / 4 / 8

Q) are these signed or unsigned? which format uses?

S) MSB reserved for "sign"

MSB = 0 +ve

MSB = 1 -ve.

byte $b_1 = 32;$

Hex $\rightarrow 0x20$

Binary: $\begin{array}{c} 0010 \\ \text{Upper nibble} \end{array} \quad \begin{array}{c} 0000 \\ \text{lower nibble} \end{array}$

byte $b_1 = -32;$

hex = $0x20$

Binary = $0010\ 0000$

~~11011111~~ \rightarrow 2's complement
 00111111

2's complement

$0010\ 0000$ \rightarrow 2's complement
~~11100000~~ \rightarrow LSB
MSB

Range -2^{n-1} to

-128 to 127

short occupies byte 2	" "	int 2	long no
2byte -2^{15} to $2^{15}-1$	-2^{31} to $2^{31}-1$	-2^{63} to $2^{63}-1$	

How to find from Java docs ~~API~~

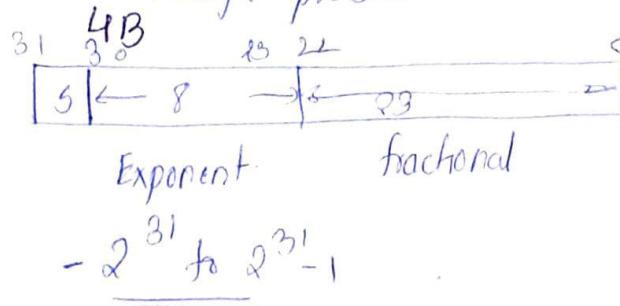
\rightarrow javadocs \rightarrow search, Byte \rightarrow min value, MAX_VALUE

class start with upper case

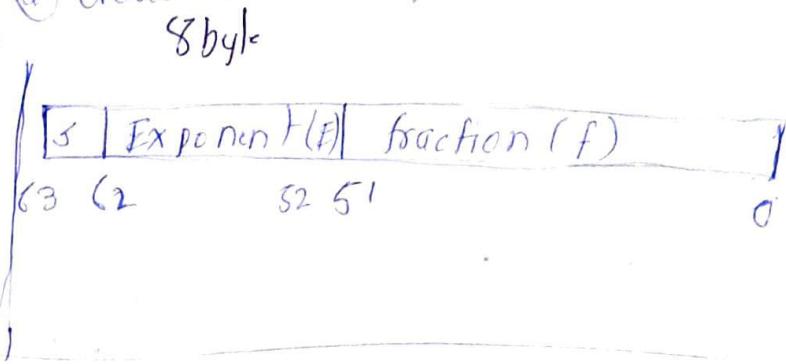
\rightarrow search box \rightarrow Integer \rightarrow MAX_VALUES, MIN_VALUES.

Floating point

① Float → single precision



② double : - double precision .



e.g. → SRC → TestPrimRange.java

class TestPrimRange {

public static void main(String[] args) {

System.out.println("Byte Range " + Byte.MIN_VALUE + ":" + Byte.MAX_VALUE);

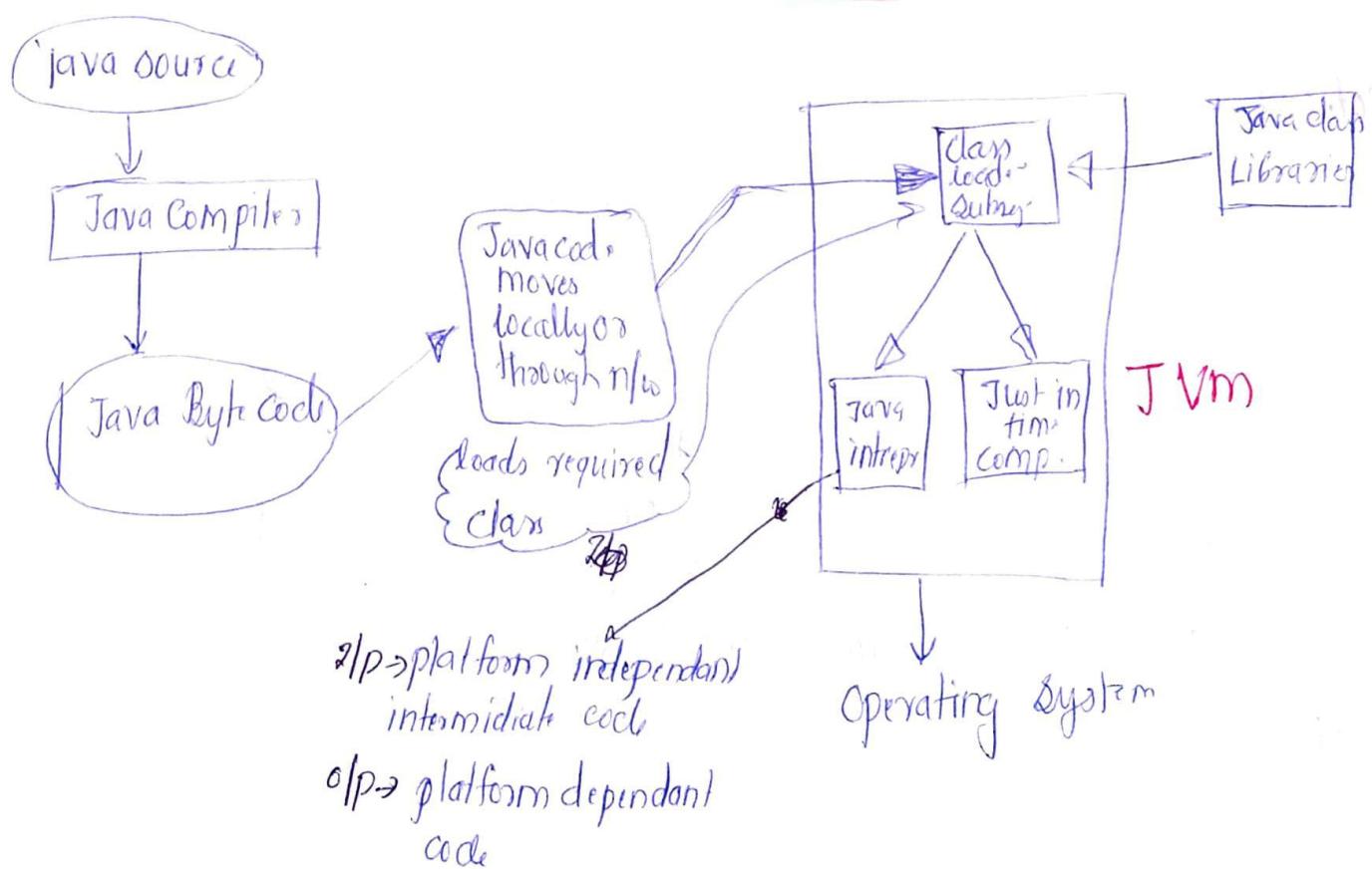
System.out.println("Long Range");

System.out.println("FLOAT Range");

System.out.println("Double Range");

SRC>javac -d ..\bin TestPrimRange.java

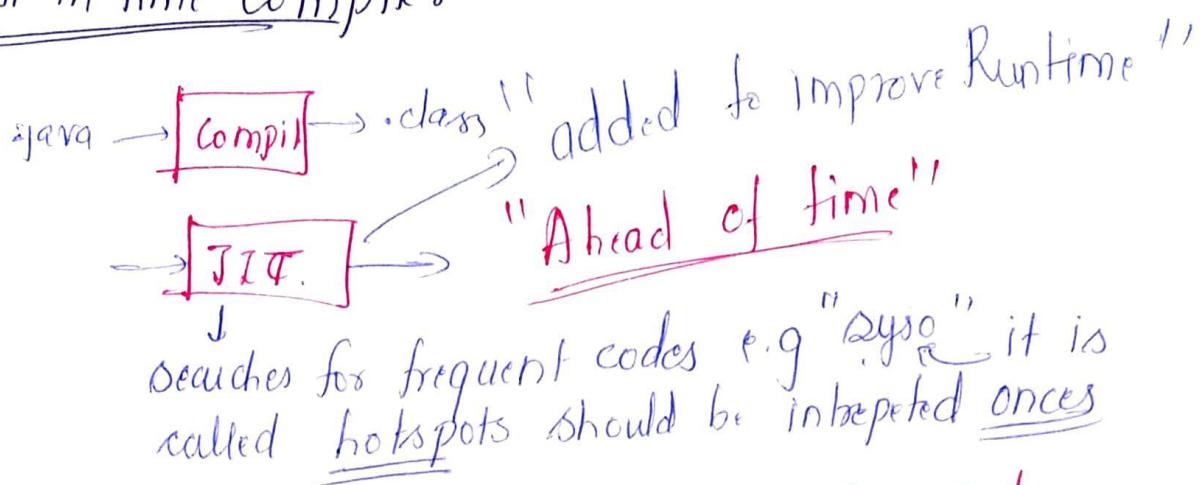
SRC>cd ..\bin TestPrimRange .

JVM Architectural View

Q) What happens if we write 100 times sys0 ?

Interpreted → 100 times

Just in time Compiler



① Hotspot profiler → JIT → Native Code

is stored by
JIT in native

② Interpreter will take it from Native Code Cache

③ next time "JIT" will code directly Machine Readable

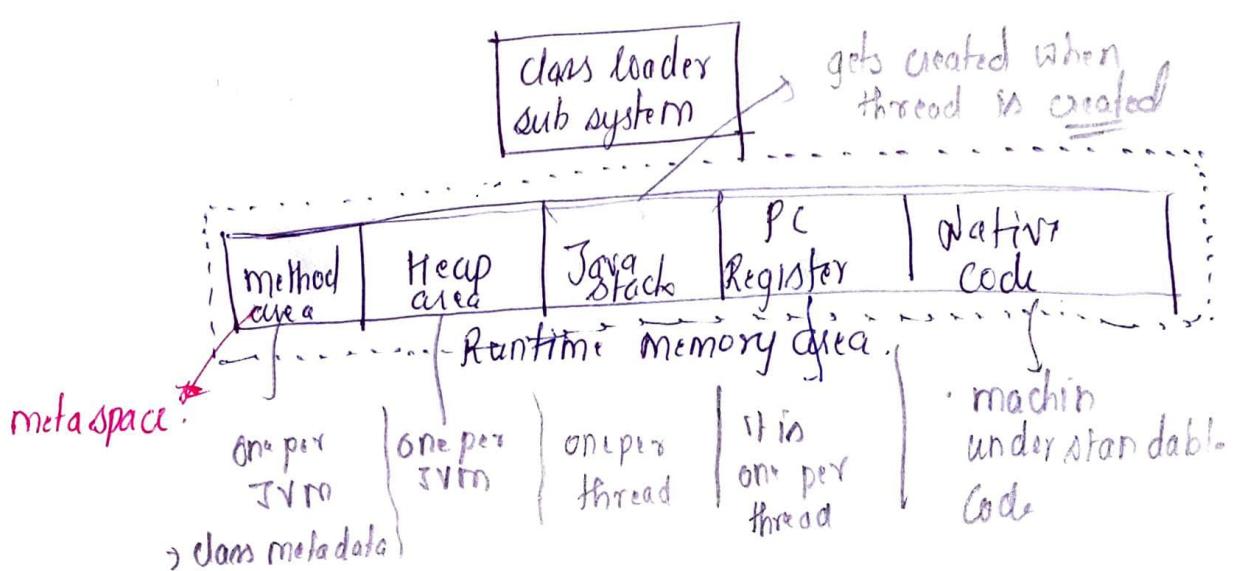
Code

Java interpreter → interprets line by line and helped with jit compiler.

Hotspot = frequently used code

Native code ≈ machine code

byte code → needs interpretation
(i.e. machine dependency)



Method space :- metadata (it is data about data)

- ⊗ loaded class information
- ⊗ single copy , static data member, constructor, /method

How many method areas are there ?

Ans : 1 per JVM

If we divide class in ten thread how many methods created .

Ans 1 .per JVM ~~not thread~~

Heap Area :- dynamically created object

- ⊗ Arrays are created in HA
- ⊗ Non-static data member

How many copies of HA ?

1 per JVM

⊗ methA and HA are shared or not shared in thread .
Shared.

Java stack :- only 1 per Java only 1 per thread

⊗ which thread created by JVM :-
main thread → static void thread

2 thread → 2 stacks created



⊗ method local info (args, local var, ref vars)

Individual stack : stack frame

main thread → looks for main method

PSVM(...)

10:10
=

{ main's stack frame

"LIFO"

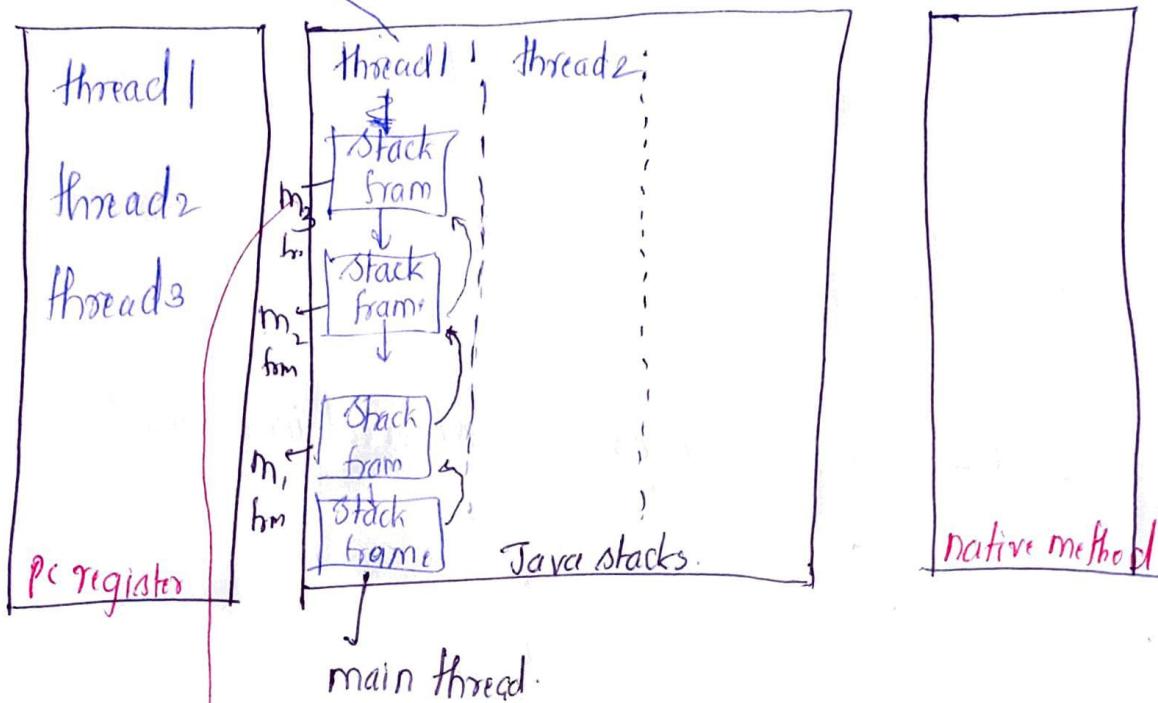
m₁()

3

m₁ --> m₂(); → m₁ calls m₂

m₂ --> m₃(); → m₂ calls m₃

main thread



How many stacks → 1 per thread

When m₃ is returned → it is poped (cleared)

When m₂ is returned → it is poped (

m₁ is returned → it is poped)

Main finally return → Stack becomes empty.

PC registers (next instrucn address)

How many PC registers?

1 per thread ~~A A A~~

Native methods: are converted to

~~per~~ ④ Java method info stored on NMS

④ are compiled directly into N.M

④ Method area is called ?

metaspace.

④ When JVM creates stack?

When thread is created

④ What is job of Main thread?

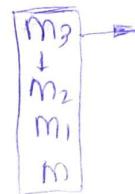
start sequential execution

will call main
this info add on stack.

④ Content of stack

local, variable, return values.

main calls: $M_1 \rightarrow M_2 \rightarrow M_3$



Java methods are compiled?

into Byte code \rightarrow they further interpreted

④ Java method info kept on java stack

④ Native \Rightarrow " into native stack.

Conversions Regarding Primitive

① automatic conversion (widening)

byte \rightarrow short \rightarrow int \rightarrow long \rightarrow float \rightarrow double

promotion possible

char \rightarrow int

~~A A~~
Inference

Rule 1: source and destination is compatible.

④ destination should be able to store more ~~data~~ than range
source

211 Why long to float ~~as~~ long \rightarrow float ~~as~~

long \rightarrow 8

float \rightarrow 4 ⚡ msB Exponent fraction

is automatic

Float_min = ?

⚡

long_min = ?

⚡ precision is compromised ~~as~~

⚡ any operatn involve in (byte, short) \rightarrow automatically to int.

① int and long \rightarrow long ~~as~~

③ long & float \rightarrow float ~~as~~

④ byte, short ... & float, & double \rightarrow double

Narrowing (forced-conversion)

non-automatic loss

① double \rightarrow int loss of precision

② float \rightarrow long

③ double \rightarrow float

TestConversions.java

```
public class Testconversion {  
    public static void main() {
```

byte b1 = 100; \rightarrow no error b/0 - 128 to 127

byte b2 = 20;

byte b3 = b1 + b2; \rightarrow compile time error

byte \rightarrow int. \rightarrow automatically converted to integer

byte * b3 = (byte) (b1 + b2)

`short s1 = 1234`

`short s2 = b1 + s1;`

~~short~~ ~~int~~ automatically convert.

`long l1 = 12345678;` → int to long

`float f1 = 12.5; // will work, not array.`
also

long → float

`long l1 = long (f1);` → not automatic conversion.

`char ch = 'A';` → no error

`ch = 65` ✓

Char to int automatic

`boolean flag = 1;` → type mismatch

→ into to boolean not possible

`double d1 = f1` → No error

→ float to double auto conversion

`d1 = 123.456(f)` → Float literal

~~double~~ ~~float~~ → automatic conversion

`float f2 = 2456.67`

~~float~~ → ~~double~~, automatic conversion not possible

Q) Short circuit operator

$b_1 + b_2 \Rightarrow b_1 = b_1 + b_2 ; \rightarrow \text{no error}$

→ if does implicit type casting

Scanner

Q) → What is scanner → it is class of scanner class

Q) I am text based parser, & can convert from any source

Scanner(System.in)

new scanner(keyboard)
(socket)

Standard I/O

of keyboard

Q) you can give console I/O, text file, socket, string

string token → integer/double/int.

Q) it has Scanner
small buffer 1k.

Q) How to read from scanner buffer. 1k.

sc.nextInt()

Internally → delimiter (separators)

white space

token

Q) token ends with space

Q) Can we change default delimiter

yes, (use delimiter)

Q) System

Q) String next() → reads till white space

only reads

SC.nextLine() → delimiter is '\n'

entire string return one token.

check type of data

F/P → 10 → sc.hasNextInt(); returns "true"

↳ just checks data type.

To read data → sc.nextInt();

hasNextByte(),

⊗ it will not remove from buffer. ↳ returns 'false'

⊗ before terminating app we should stop scanner
↳ leaky resource

close();

sc.close();

⊗ System.in →

Precision 3digit

(%.20.3f)

Boolean → %b

To add new line

"%n"

buf.append("\n")

Day 2 OBJECT ORIENTED PROGRAMMING

Class: blueprint for object
① it has properties (data), methods
regarding class in objed.txt

Class declaratⁿ:
① Access specifier: public / default
② Class Name: begin with capital letter & then Camel Notation.

(3) **Super class**: parent class, base class.

What is universal parent class?
Object

④ 
A A O
B B

⑤ Interface 

⑥ Box = { }

best way to initialise state of object:

⑦ Constructor

OBJECT

- ⑧ State: stored on heap,
- ⑨ Behaviour: action on data.
- ⑩ Identity: Emp id, Student PRN

Creating Object

- new → Runtime operator
- Allocating memory →

Default value

12:26 24

char :-

'New' operator

float : 0.0

↳ returns reference to that memory.

double : 0.0

Constructor :- name same as class name

⊗ no explicit return type

⊗ it has inbuilt constructors in java

Default constructor :- 0, X, 0

TD, name, salary

⊗ Null, 0.0

Default initialised value for reference type is Null

⊗ Parameterised constructor

⊗ "this" → to resolve conflict ambiguity

this.name = name;

this → represent current object reference.

⊗ can we add overloaded object

create a class to create 3D box; (12:50)

right encapsulation :- make all non-static members privateDefault → package restricted

21x
Box.java

Package Private → Default
↳ Public

class Box {

// Data members state, instance variable
// tight encapsulation.

private double width, depth, height;
// constructor parameterised.

Box (double width, double depth, double height) {

this.^{width} = width;

this.^{depth} = ~~width~~ depth;

this.height = height;

is optional

}

Behaviour → instance level, restricted to object

String getBoxDimensions() {

return "Box Dims " + this.width + "
+ this.depth + " " + this.height);

String + double,
" " + int

}

To Return computed volume.

double getBoxVolume() {

return (width * depth * height);

* TestBox.java (import java.util.Scanner;

class TestBox {

pu. s. V. m() {

Scanner sc = new Scanner(System.in);

sys("Enter Box Dims: w'd'h");

Box bi; → reference type Variable of type
class

Reference
Variable

Box b1; "Method local variable stored in stack"; frame",

How many byte allocated = JVM specific ~~depends on system~~

System
`= new Box (sc.nextInt());`

Box b1; → box type reference

b1 = new Box (); → new keyword returns address.

Object creation.

Display dimension

`System.out.println(b1.getBoxDimen());`

Display Volume

`System.out.println("Volum: " + b1.getBoxVol());`

Use a class Relatn.

Memory

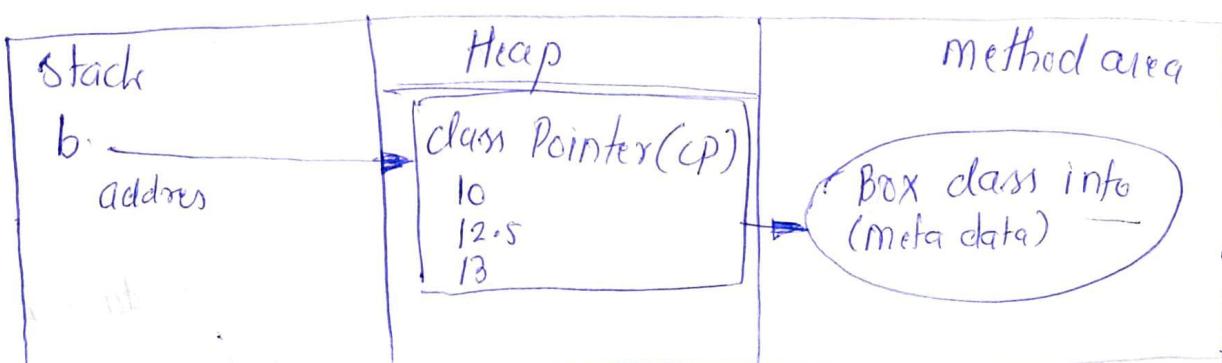
Box b1; → reference creation

b1 = new Box (sc.nextInt(), sc.nextInt(), sc.nextInt()); // 10 12.5 13.

`System.out.println(b1.getBoxDimensions());`

RHS execution

`System.out.println("Volum: " + b1.getBoxVolume());`

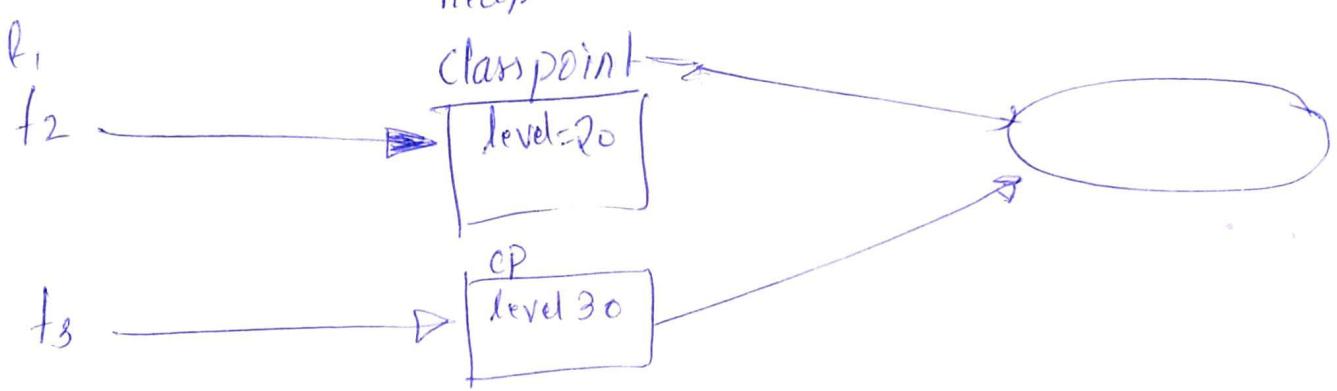
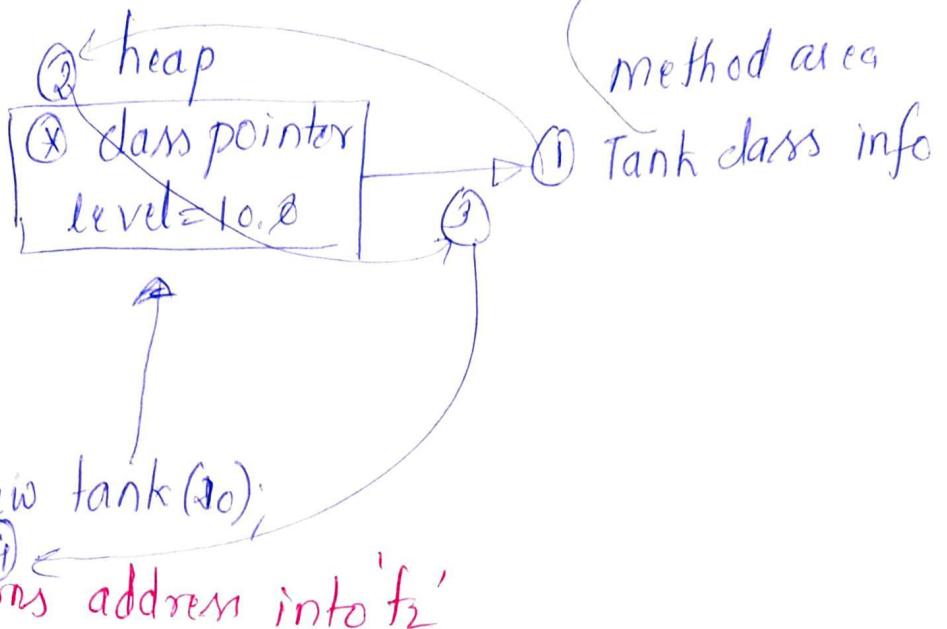


4:00 Java session

⑧ assignment Tank assg 2

stack

+↑



19/10/2021 8:19

Day 3

⑨ class loader :- locate'n' load required classes in the JVM's memory.

⑩ Run time data areas (memory mod.).

Method areas :- shared resources / class meta data + allotted mem for static data members, metaspace directly allocated from comp's native memory

③ Stack is created one / method invocation. 220

How many stacks? / when created?

2

Stack created by JVM

① main thread → initially empty.

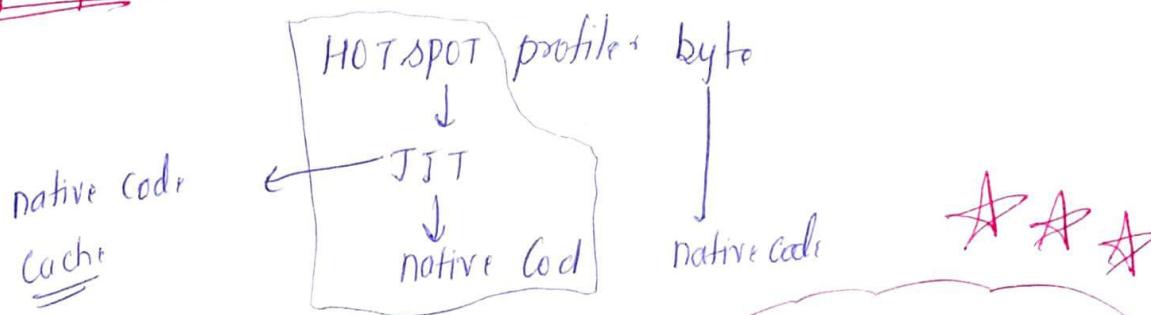
② Garbage collector

Meth. stack frames :- method local info (eg: args, local vars, return values)

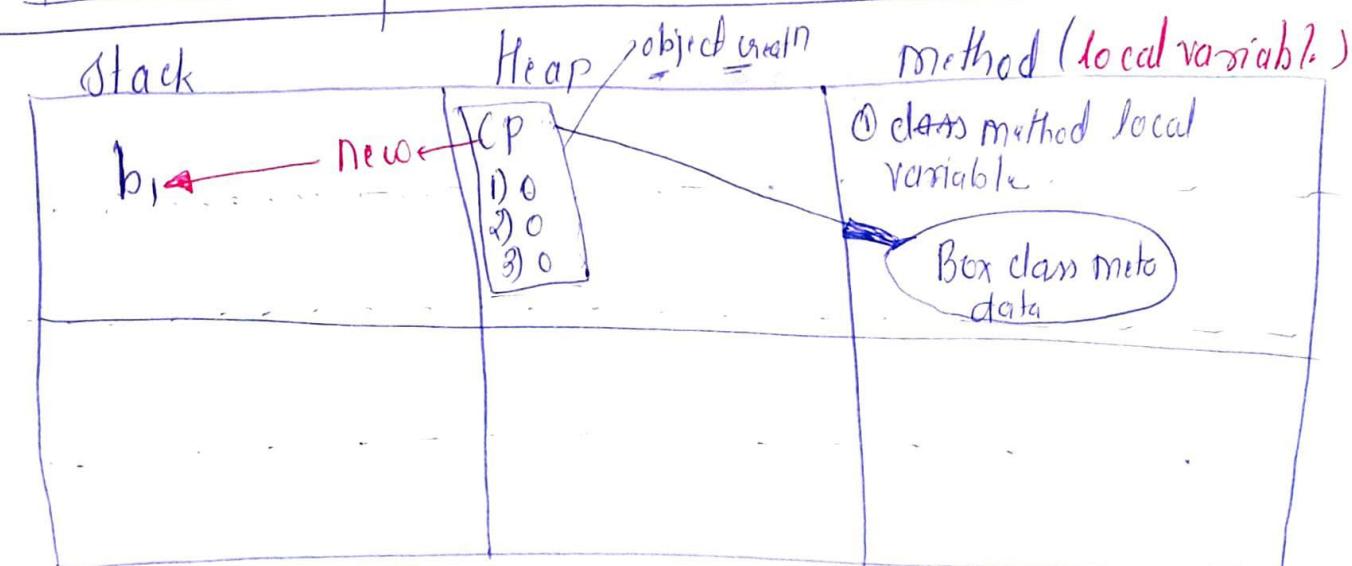
What is native code (if third)

③ code which is directly converted to machine level :-

Hotspot profiler: looks for hotspots



④ If	80% code repeated Interpretation	20% new JIT
------	--	-------------------



~~(*)~~ one argument is passed by default is this

⑧ implicit argument is 'this' passes - .

explicit argument = 0 implicit argument - (1)

this

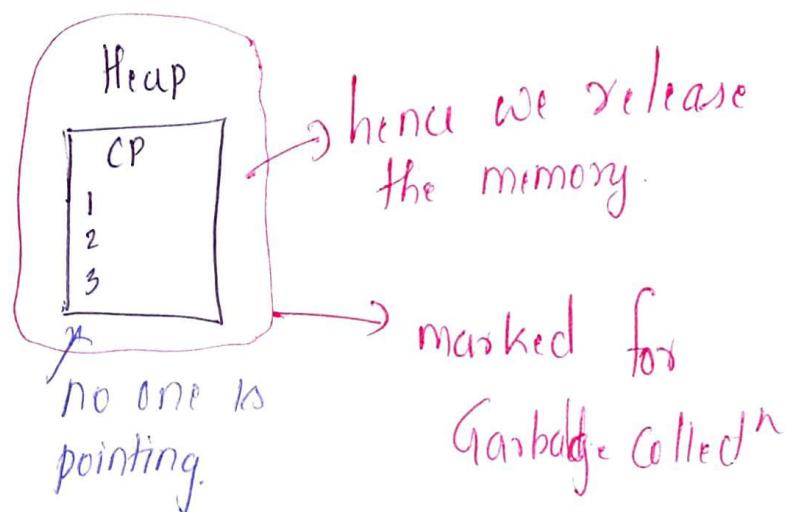
Dereferencing :- b1=null;

Null pointer Exception → will abort execution.

Stack

b1=null

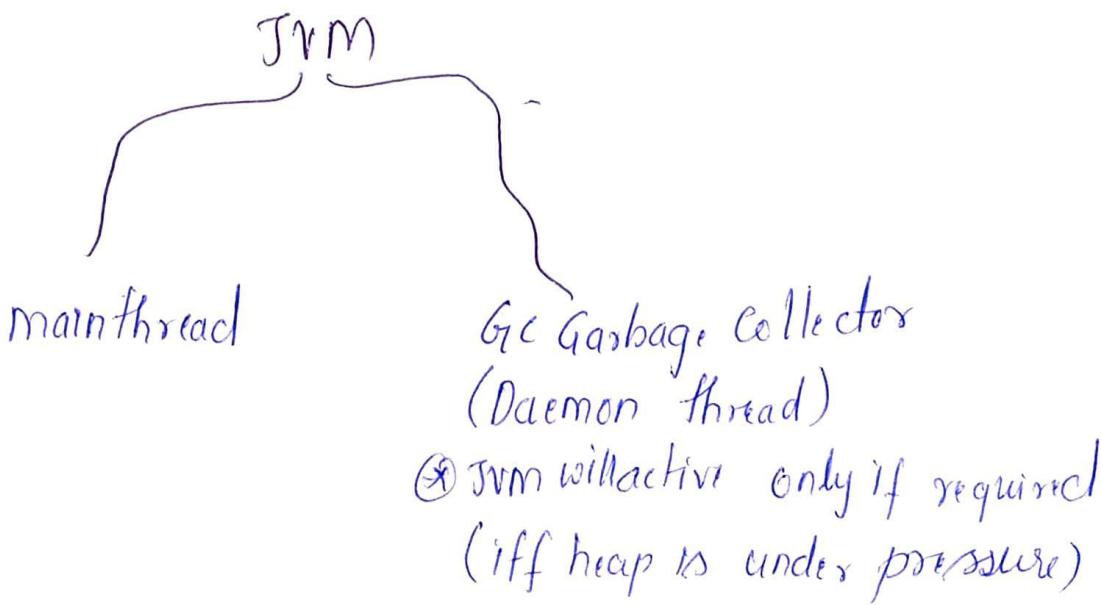
b2=null



process of Garbage Collection

⑧ Garbage = unreferenced object

⑧ To avoid memory leaks/hole.



9:23

⊗ Garbage collector search only in heap.

⊗ Garbage Collection is slow process

flow to request Garbage collectⁿ?

public void static void gc();

it is request/order → Request

Can we request GC for Garbage collectⁿ?
Yes not order.

protected void finalize() throw throwable

⊗ Garbage collectⁿ → needs to stoped the world.

System.gc() →

AJD Interview Question

are there any candidate marked for GC?

Box b1 = new Box(10,20,30); Yes 1.

b1 = new Box(3,4,5);

b1 → CP
10,20,30

→ CP
3,4,5

AJD interview

What are the triggers for GC?

b1 → CP
123 object marked
for null

⊗ Nullifying all valid
① ref trigger

(2) Trigger Box $b_1 = \text{new box}(10, 20, 30);$

$b_1 = \text{new box}(1, 2, 3)$

* Re assigning the object *

Second trigger.

What happens to method area?

Q Meaning of heap under pressure 2

If there is no space in heap to create more object,
JVM activates the GC thread, But it refers to heap only.

Pointers Vs references in Java

Similarity:- pointer & reference : holds address to the object created in heap

Difference:- To add robustness, pointer arithmetic not allowed.

Right associativity

$$\text{width} = \text{height} = \underbrace{\text{depth}}_{\textcircled{1}} = \underbrace{\text{size}}_{\textcircled{2}}$$

(3)

Reusability One Constructor call another Constr.

Constructor Chaining

1

Box(double side) {

10:19

this usage this(); → call to default

(calling
constructor)

④ When we add user constructor default constructor
disabled:

 this(side, side, side);

~~DRY~~

do not repeat yourself

AAA

One line ⊆ Optimized

④ Box b=new Box(); → for no error

Box() {

 this.width = -1
 this.height = -1
 this.length = -1

 this(-1, -1, -1);

AAA

 this(-1);

this should be first line

Box (double side)

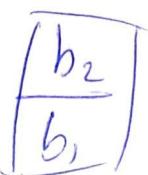
Box(double, double, double)

Passing Values

System. `sys()` `b1`, `isequal(b2)`)

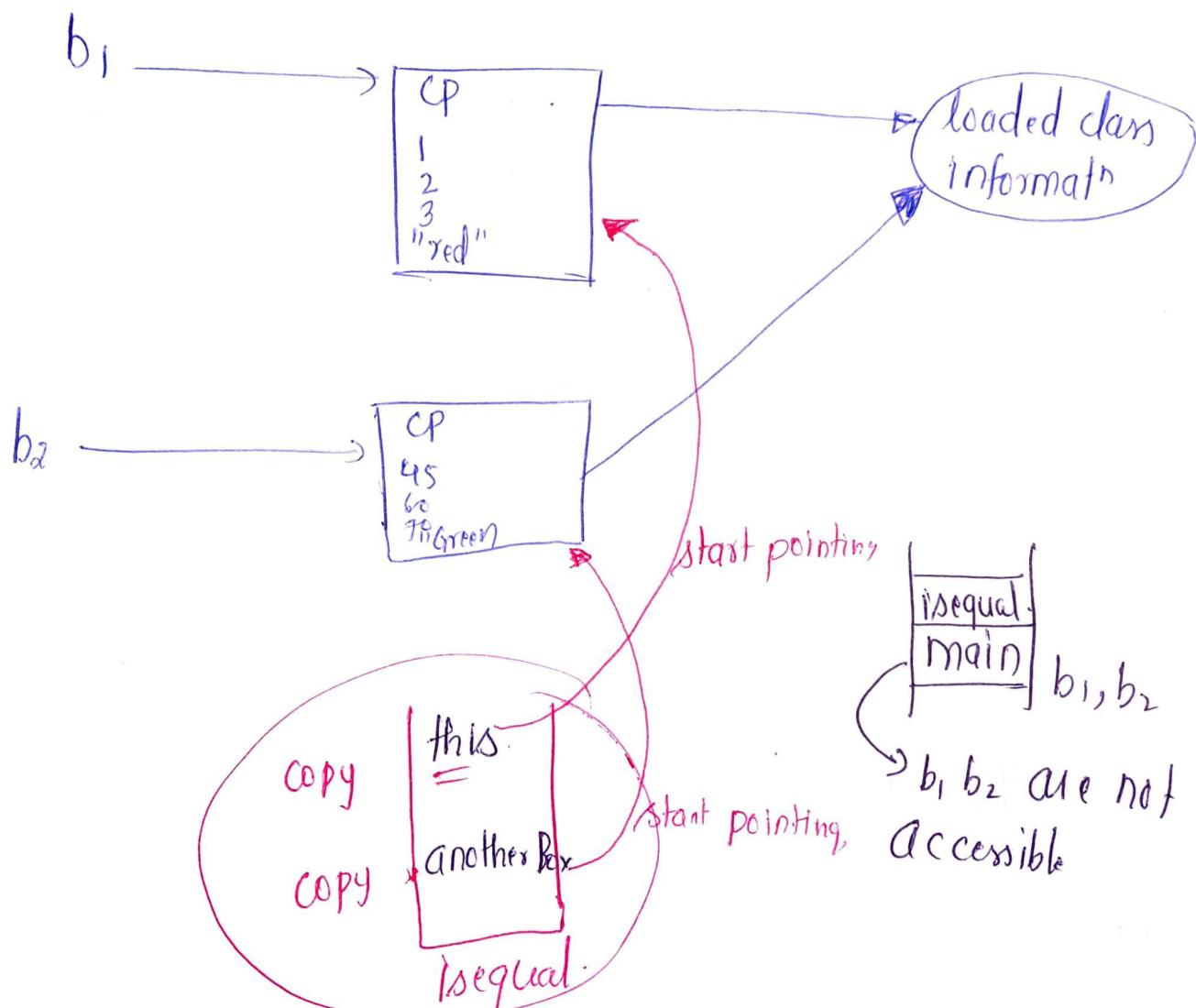
Passed by copy.

↑ reference type and



`this = b1`

`anotherBox = b2` (copy of reference)



when popped X

How many objects are marked for GC? after returns.

X because

O because copies were passed

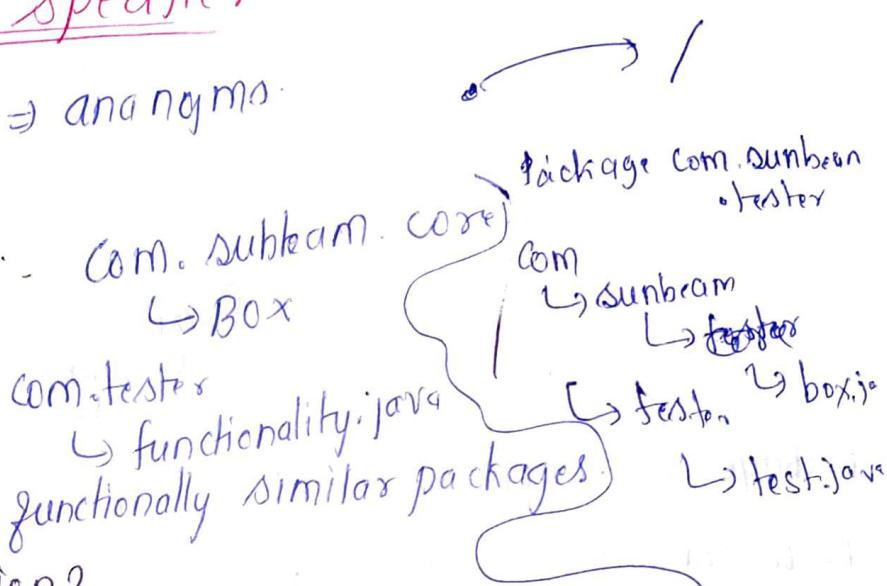
whether primitive or non-primitive everything passed as copy 12:00 226

`this.hashCode();`

Package and Access specifier

Default value of package \Rightarrow anonymous.

Reverse Company name :-



④ package is collection of functionally similar packages.

What is name space collision?

allowing duplicate class names placed in different packages

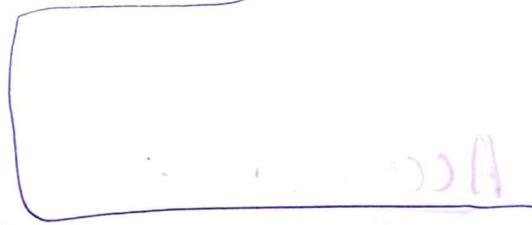
④ Package \Rightarrow has to be placed as the first statement in java source.

package p1; \Rightarrow the classes will be part of package p1;

④ all lower case.

package p1; class A();

A must exist in folder p1.



④ is it compulsory

if it is recommendation.

④ fully qualified Name.

java p1.A

~~import~~ ⁿ import package.class.

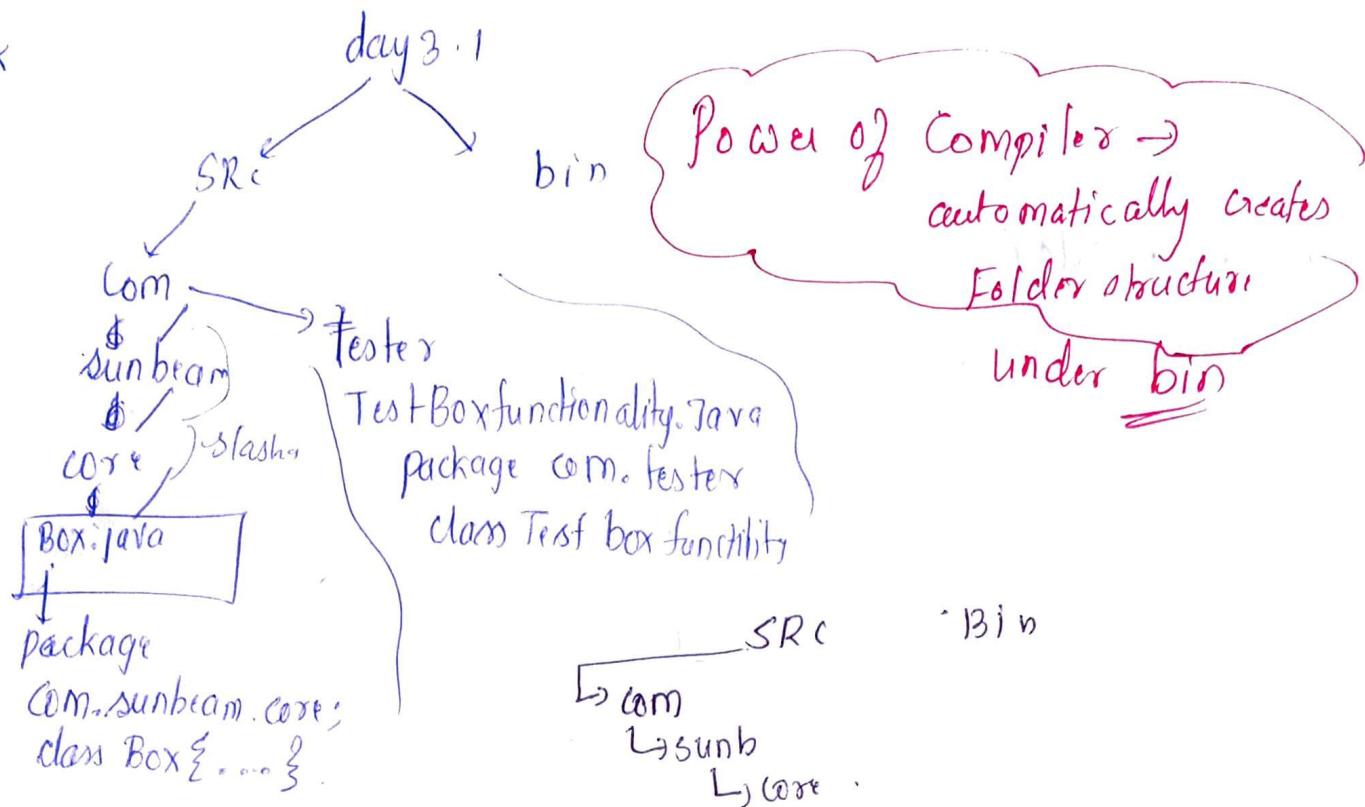
com.sunbeam.core.test.java;

④ new folder \rightarrow SRC \rightarrow com \rightarrow make constructor, methods

BIN

public
keep data members private

day 3.1



Package `com.sunbeam.core`; → always first statement.

Now compiling file from package.

`src>javac -d ..\bin com\tester\sunbeam\com\sunbeam\testboxfunctionality`

Error → symbol and class not found error

Solution

```
package com.tester;
import java.util.*;
com.sunbeam.core.Box
```

Access specifier → Public → make it or
Default → you cannot access
in other folder

④ after compilation automatically bin file is created

④ for compilation

`bin>java com.tester.TestBoxfunctionality`.

④ for accessing through package compulsory

fully qualified name: `com.tester.TestBoxfunctionality`,

→ must match with folder structure or it will
be error

⑧ every time we will have to go the bin folder and write parent hierarchy :-

hectic every time going to bin.

without ".class" → program won't work.

com. tester. TestBoxFunctionality. class
~~it will check for package. if .class not~~

To make it run from any folder :-

classpath :- To run package fro class from any package
 we must set specific environment Variable.

13:36

set class path = g:\dac\day2\bin;

How to set environment variable.

Variable Name :- classpath root

3.1\bin; → current folder

madhur> set classpath;

madhura> java com.tester. TestBoxfunctionality

cmd

src> set classpath ⇒ path not set.

src> javac -d ..\bin com.tester. TestBoxFunctionality.java

bin> cd.. \src

src> java com.tester. TestBoxfunct. → error.

goto bin

bin> java com.tester. TestBoxF → fully qualified Name

bin>

229 Variable name :- classpath

Value :- Top up package hier.

= day 3-1 > bin

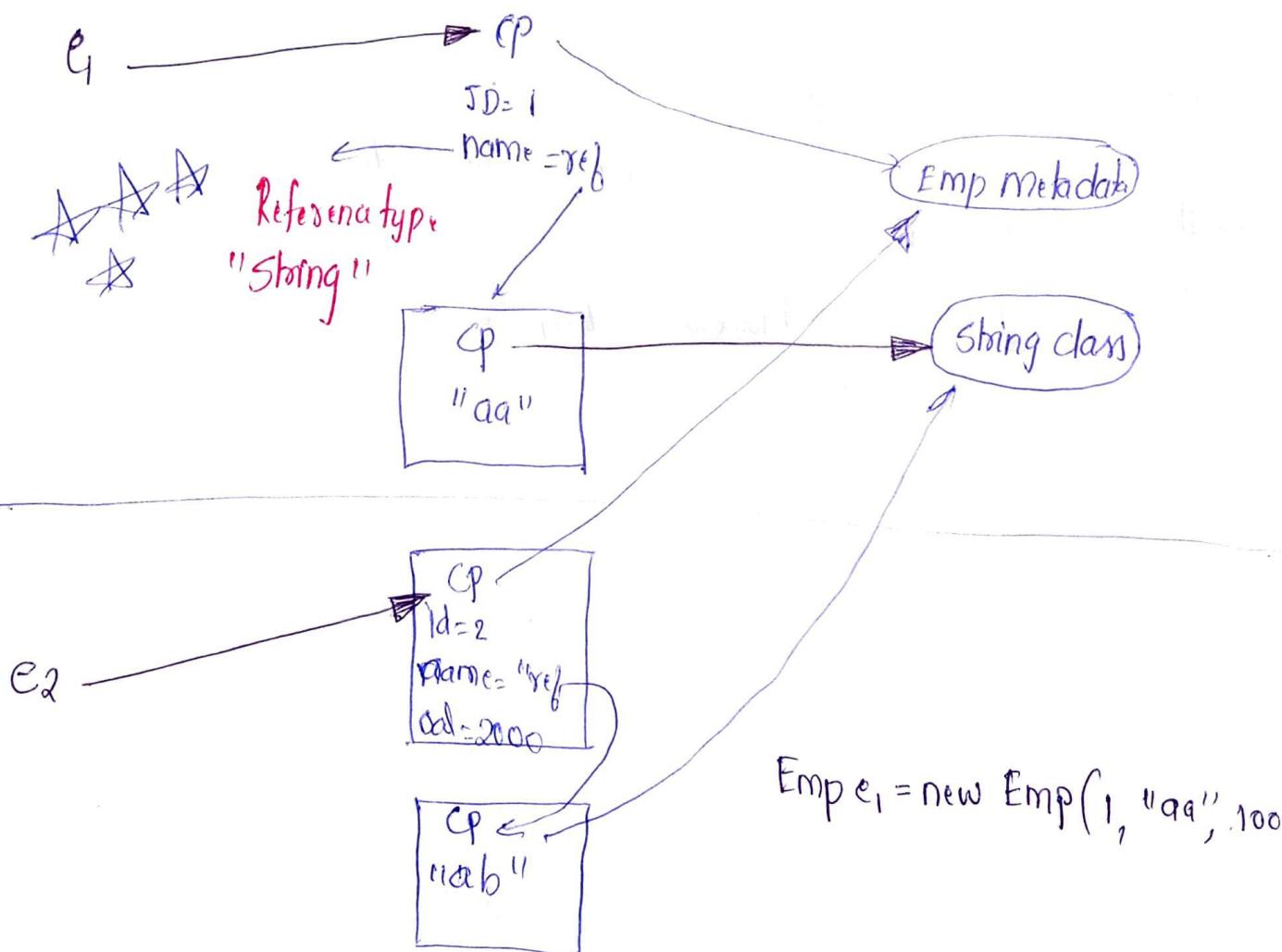
"D:\sunbeam\bin"; → ^{delimiter} for current folder,

Open new cmd prompt:

Users>madhura> set classpath
VV

madhura> Java com.tester.TestBox functionality.

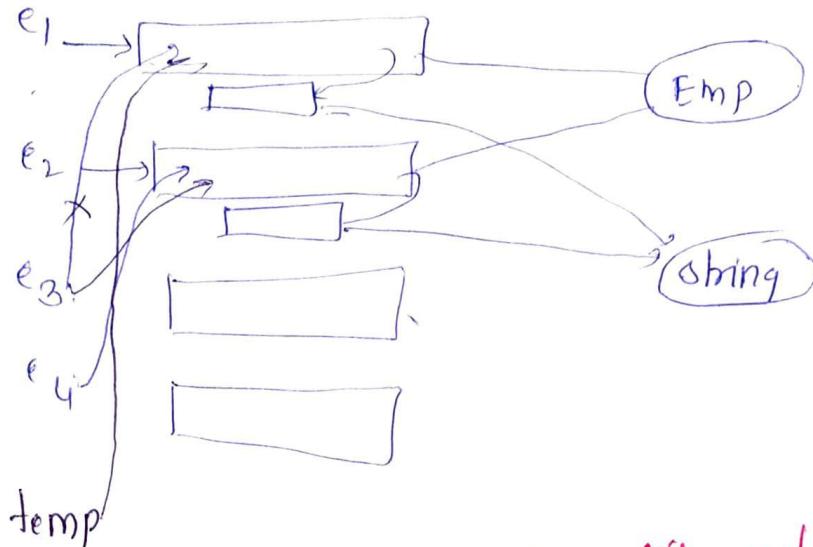
20/10/2021 Day-4 :- Storage of String(ref)



$\text{swap}(\text{Emp } e_3, \text{Emp } e_4)$:

copy of reference "and copies are passed"

$e_3 = e_1, e_4 = e_2$



$\text{swap}(\text{Emp } e_3, \text{Emp } e_4)$

{

$\text{temp} = e_1;$

$e_3 = e_4$

$e_4 = \text{temp}$

After return of swap

⊗ e_3, e_4, temp is gone.

⊗ no candidate for Garbage Collection

⊗ we will get original salaries

Access Specifier:

	Same class	Same package Sub class	Same package	Different Packages	Different Packages
Private	Yes	No			
Public					
default	Yes				
protected	Yes				
Public	Yes				

Visibility

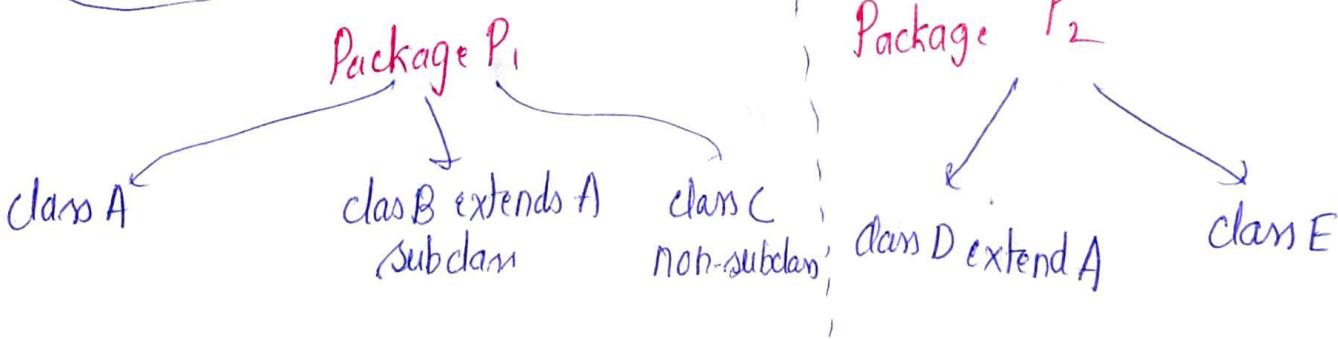
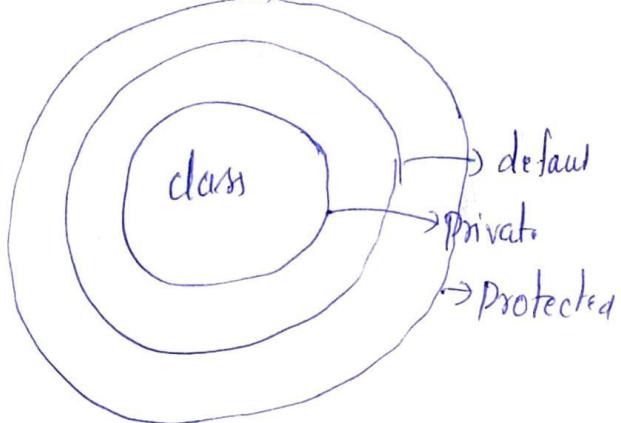
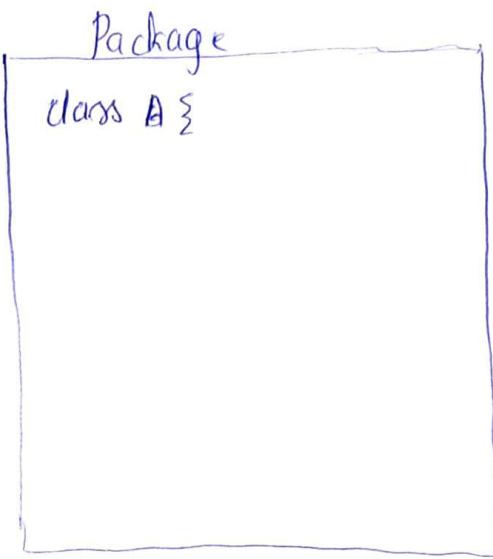
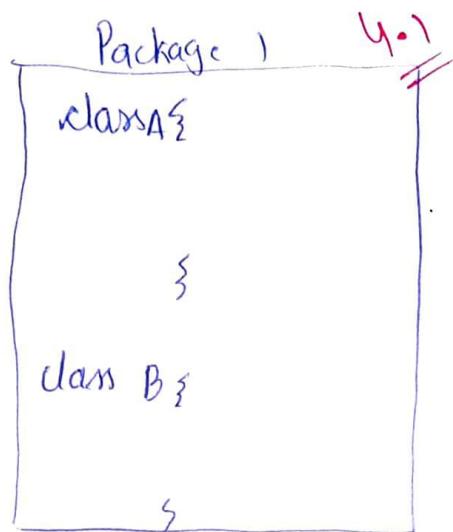
Same package Sub class

73) Private :- same class.

default : same package.

Protected : different package subclasse (class in other package)

Public : every where



Package 1

class A {
private int i;
int j; default
protected k; no compiler
public int l; error!
A() {
 System.out.println("i " + "j " + "k " + "l");
}

Sub class in Package 1

class B extends A {
 B() {
 System.out.println("i " + "j " + "k " + "l");
 }
}

System.out.println("i " + "j " + "k " + "l");
} *not accessible*

Part non sub class in Package 1

class C {

```
(())
{
    sys0( "i", "j", "k", "l");
}
private protected
```

Protected is accessible
in same class

class C {

```
A a1 = new A();
```

```
(())
{
    sys0( "a1.i", "a1.j", "a1.k", "a1.l");
}
```

★ ★ in

protected = default in same package

Package 2

import P1.A;

class D extends A {

this will also
not work
till we make A
public

D() {

super();

```
(())
{
    sys0(i, j, k, l);
}
private default protected public
```

① if we don't add constructor
compiler adds default constructor

② invoke immediate super class

constructors Protected / Public

③ Make A() → public A()
then only parent constructor called.

Package 2

~~import P1.A;~~

class E {

E() {

A a1 = new A();

sys0(a1.i, a1.j)

x. a1.k

protected public .

change from protected

to public Constructor of class

Arrays :-

- ⑧ always an full fledged object (all rules of object are applicable).
- Java does → object
- ⑧ they are created ^{→ root of the class hierarchy} Runtime
- ⑧ always are associated with class.
- ⑧ ~~All All All~~ all arrays with same dimension and type have the same class

[I]

length → Data member:

- ~~All All~~ array size is fixed "Immutable";
~~All All~~ implicit super class is Object

Create a array of primitive type:

(ctrl + shift + W → close all)

package → Prim_array.
→ add main method.

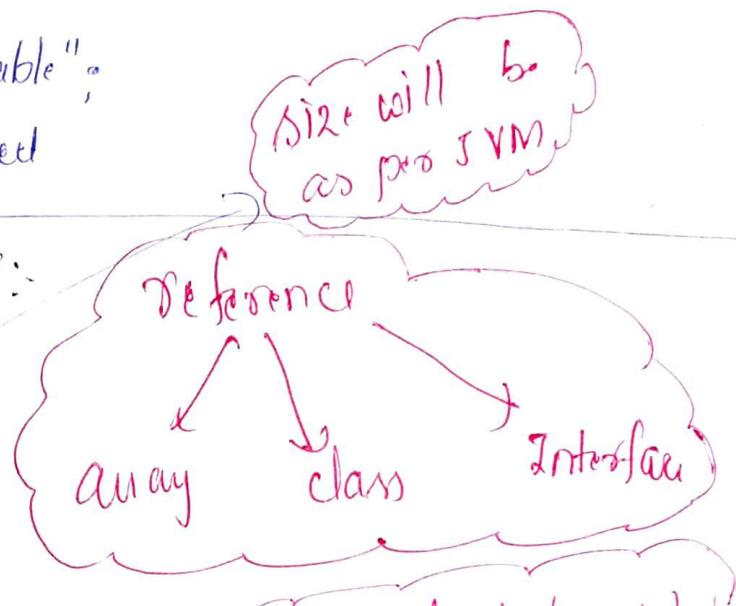
main() {

Scanner sc = new Scanner(System.in);
System.out.println("no of data sample");

⑧ double [] data;

data = new double[size]; Capacity.

for array typ.



Method local Variabl.
↳ allocated on stack
→ zero object are created.

⑧ data is ~~ref~~ reference type

⑧ data can refer to any array holding double type of values.

other wise we write constructor

What will be name of class loaded in Method area for array.

To see the class loaded in method Area

System.out.println("Name of class loaded for array " + data.getClass());

System.out.println("Default contents of array");

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

System.out.println(data[i]);

Name of class [D] → fixed for all sizes.

array class are compiler generated (hence no java docs)

{ hashCode, Object }

Inherited from Object class

Memory Picture of Array

double[] data → no object creation.

data = new double[sc.nextInt()];

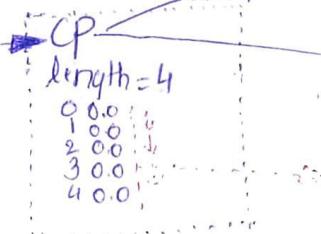
① data is reference holding addr.

① class loading
② array obj.

Stack

data

Heap



Memory Area

① Array class loaded

[D]

What is data member of array?
length

23 ~~for~~ for each

④ just iteration in requirement

double[] data = new double[10];

for (double d : data);
each ↑
silent loop local
variable in

Syntax

reference Variable.
name.

① Loop → d = data[0] → Copied for size 10

② → d = data[1] → Copied d = data[9]

3 → d = data[2] → Copied =

~~for~~ for each works on copy of array elements.

for dynamic z/p

? last element

for (double d : data) → {data.length - 1}

limitation

④ can only go from 1st to last with step size +1

④ can't be used for modifying elements.

Comment → d + 1

~~for~~
=

for (double a : data);

d = d * 2;

POC of for each

modifying for each content

10:23

④ single dimension array [D] class called.

Built in API for Printing array

Java docs → class Arrays

④ Various method to manipulate array.

④ To

ToString to String

toString (boolean[] a)
 ↗
 ↗ may be different
 ↗ method overloading.

O/P → string representation of the contents of specified array.

public static String toString (double[] a);

How to use

System.out.println (data);

array of References

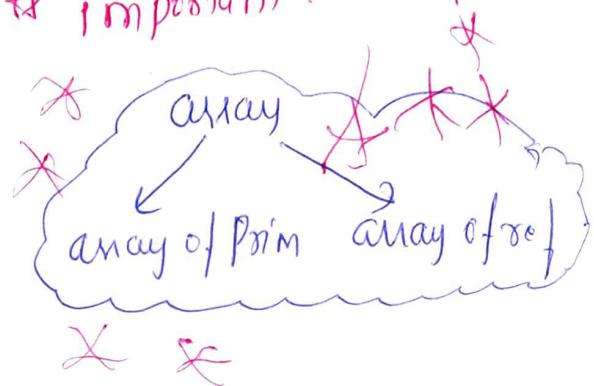
★ important

① add Box class into com.sunbeam.core.

- ① Accept box dims
- ② store them

③ Display via for each

System.out.println ("How many boxes");



Box[] boxes; ← array reference created

boxes = new Box[sc.nextInt()];
 for (Box b : boxes)
 for (i=0; i < boxes.length; i++) {
 System.out.println (boxes[i]);
 }

Only holder to keep
Box 10 done
array obj

O/P 5 nulls, size=5

M. Picture

H.o.A

M.o.A

Stack
boxes
↑ reference

CP
length = 4
0 null
1 null
2 null
3 null
4 null

default value
null for ref type

L com.sunbeam.com

Accept box dimension

only '2' types arrays →

(*) We will create object of box and wrap dimension.

① Create object and store it in boxes arrays.

for(i=0 ; i< length ; i++)

{

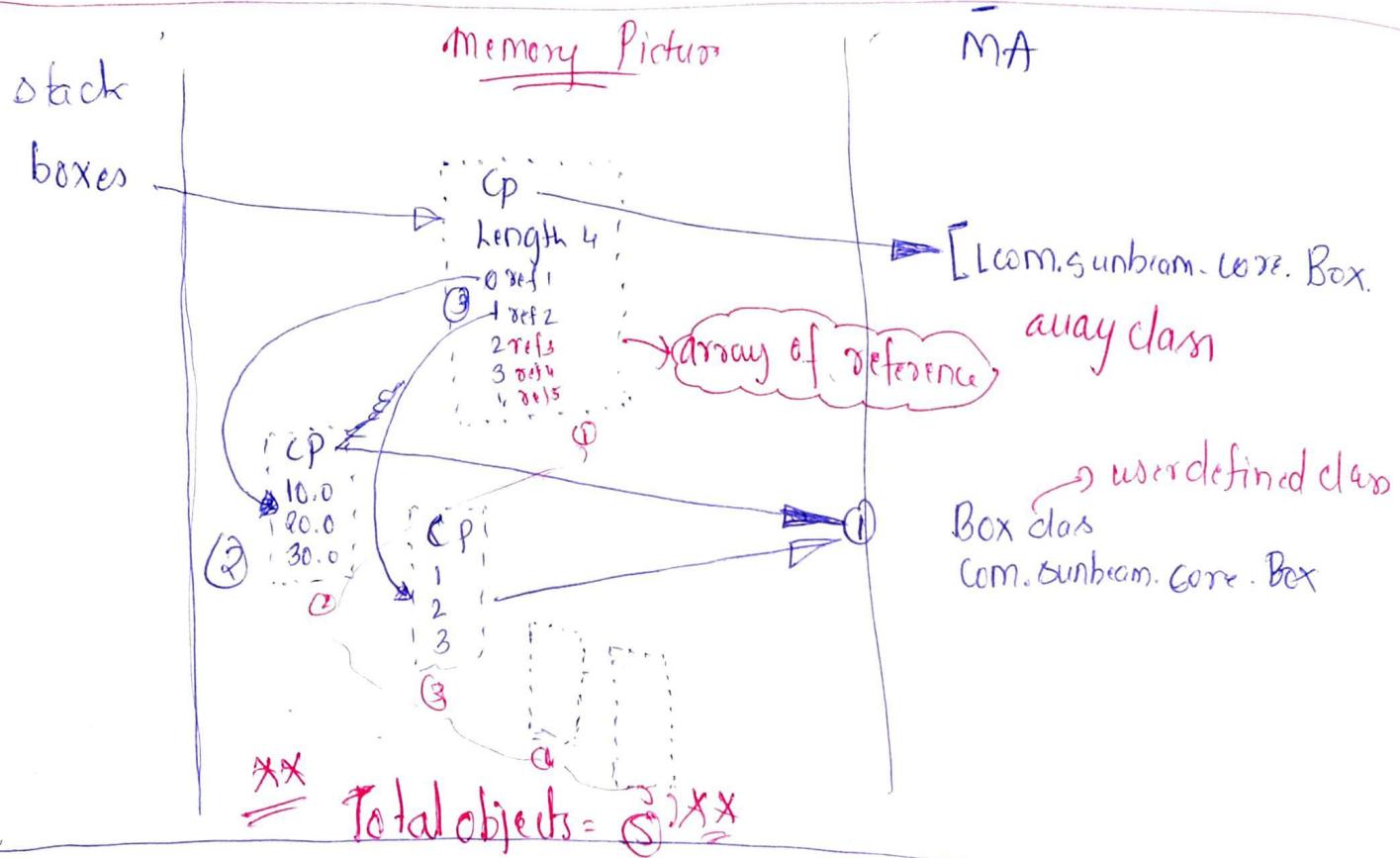
sop("enter box dims")

10	20	30
1	2	3
3	4	5
1	2	3

② boxes[6] = new Box(~~double~~ sc.double(), sc.double())

① Box created

→ wrapping dims



main() {

Scanner

syst("enter no of boxes")

Box[] boxes; → array type of ref ⇒ boxes can refer to

boxes = new Box[sc.nextInt()]; an array containing refs of Box

syst("array.toString(boxes)");

Print class loaded. →

9 nulls are sitting if 9 Boxes.

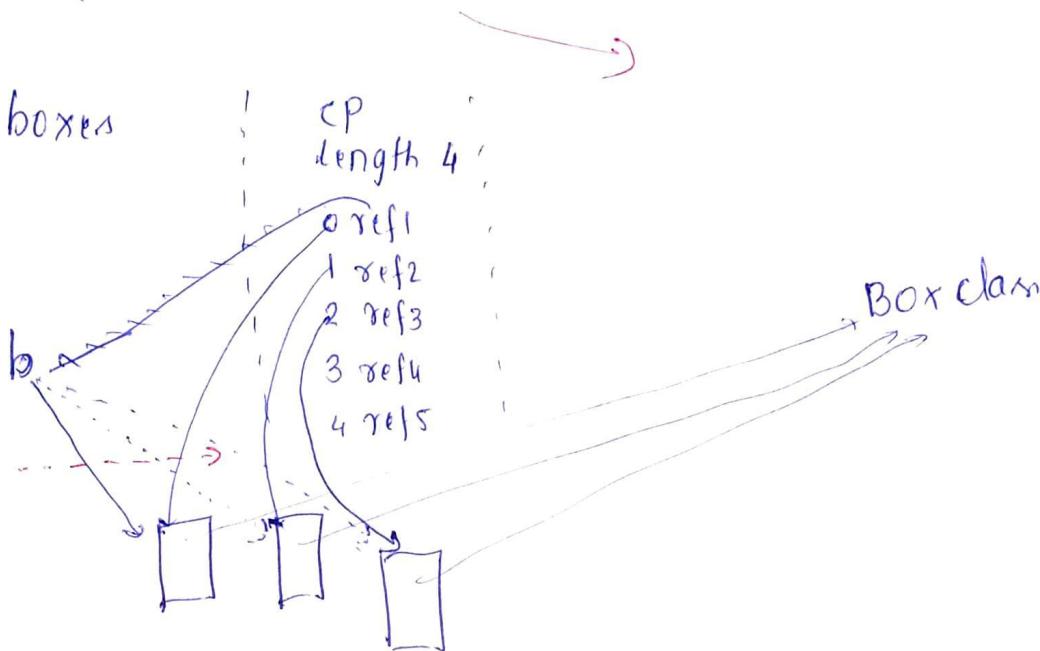
[Lcom.sunbeam.core.Box]

Take box dimensions and wrap into box reference

```
for (int i=0; i<boxes.length; i++) {
    System.out.println("Enter box dims : W D H");
    boxes[i] = new Box(sc.nextInt(), sc.nextInt(), sc.nextInt());
}
```

Display using for-each

```
for (double Box b : boxes) // b=boxes[0], b=boxes[1] . . . . .
```



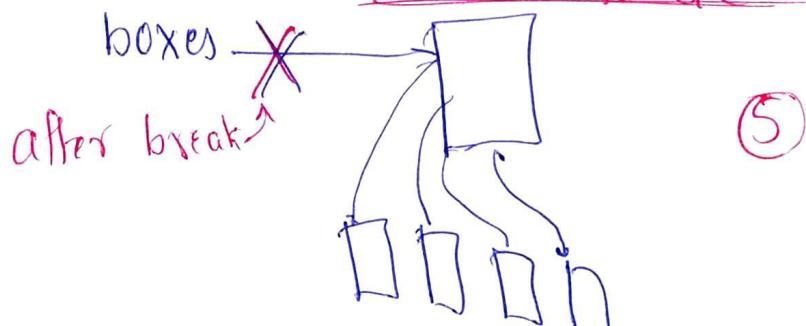
2) for (Box b : boxes) {

```
System.out.println(b.getBoxDims() + "Volume" + b.getBoxVolume());
```

* after each goes out b is removed

```
System.out.println(Arrays.toString(boxes)); → op [com.sun.proxy.$Proxy1@123,..,1]
```

Candidate for GC



23) Double the box width $\underline{\text{vol}} \geq 100$ using (for: each)
100% possible to change using for: each

for (Box b : boxes):

if (b.getVolume() > 100) {

b.~~width~~setter(b.getVolume * 2)

}

12:52

display();

}

S static

① static is key word.

② memory allocated only once @ class loading.

③ not saved in HEAP

④ stored in Method area

Rule :- How many copy you want

if 1 → static

100 → non static

⑤ initialised to Default values.

int o double 0.0 boolean false ref = null

Flow to call

class Name.memberName;

~~* *~~ Static method → can be called w/o instantiation

cannot access "this" or "Super" from within static method.

implicit parameter: "0" no this

Access of Static

④ static can access other static members directly. (w/o inst)

→ Yes

⑤ Can static method access non static members directly?

→ No

⑥ non static can access static member directly?

→ Yes.

main () {
// add instance variable.

 private int i;

// add static variable;

 static private static int j;

// add static method

 Public static void show () {

 System.out.println("non static " + i); → error X in non static.

How to access static variables

⑦ using object; it is possible.

Test t1 = new Test();
t1.show();

System.out.println(this); →

Error Can not use this

Q4) ~~(*)~~ Super is also not allowed in Static.

Who is invoker of main Method

~~(*)~~ main thread.

int 10/0 → It shows how main is called.

public void println() {

 all members are accessible

}

Static can also come with import

static import; → we can access all static members
of class.

import static java.lang.System.out;

main()

{
 System.out.println();
 out.println();

→ Import static java.lang.System*

import static java.lang.Math.*

import java.util.Scanner;

Scanner sc = new Scanner(in)

13:32

gc()
exit(0)

System.exit() → App terminate

System.gc() →

★ ★ ★ "Static initialiser Block" ★ ★ ★

242

- ④ They appear within class and can access only static members directly.
- ④ can have more than one block but not recommended
- ④ will be called once @ class loading time by JVM loader

Best way to initialise static members :-

static init Block.

- ④ that block is called only once we keep in static init block.

21/10/2021 D-5

[L → represent reference type]

- ④ for initialisation it is compulsory to use "for" "for-each" doesn't work.

④ Dynamic Initialisation of array :- of primitive type

int[] ints = {10, 20, 30, 40, 50}.

★ ★ ★

→ Implicitly added by java

int[] ints = new int[]

Memory picture

ints

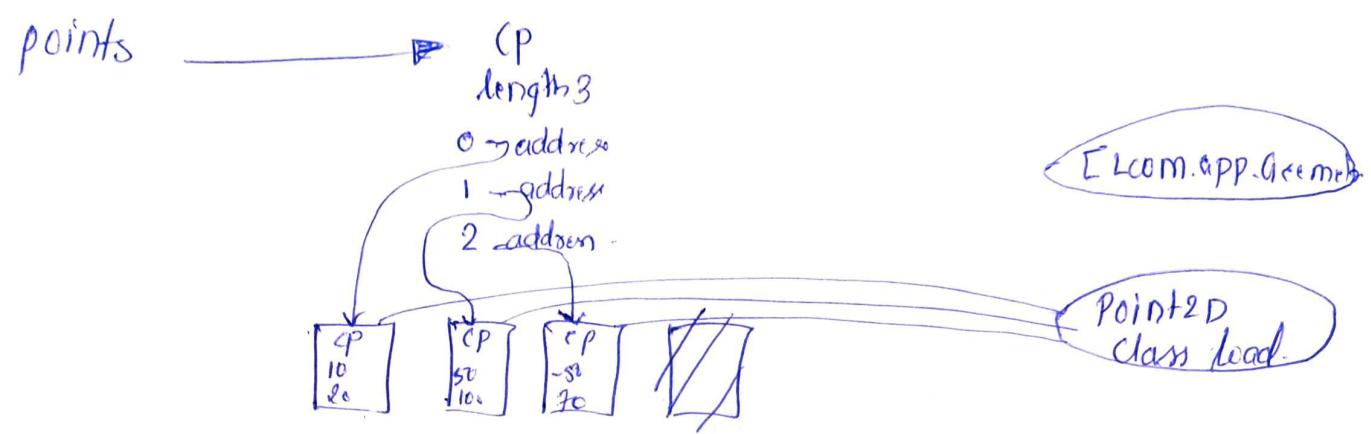
```
cp  
length=5  
0 10  
1 20  
2 30  
3 40  
4 50  
5
```

[I class info]

~~113~~ Interview tip
always fake test input and don't take from user.

Easy way to develop sample data

Point2D[] points = {new Point2D(10, 20), new Point2D(50, 100), new Point2D(70, 70)}



Java.util.Array
package class:
array.toString(ints);

④ requirement :- `sop(Value of P1 + P2);`

import → To import class from particular package

Import static → To import static method

→ Import, static java.lang.math.PI;

for all → * → applicable to static method only.

Instance Variables : (object variables)

④ Static Initialiser block :-

④ non static → constructor

④ static → static Initialiser block

④ can add functionality which has to be called precisely once.

Non-static initialiser block (Instance Initialiser)

244

③ appears within class.

{

will be called per instantiation → before matching constructor

Better alternative → parameterised constructor

}

Static nested class

class outer {

 // static and non-static
 static class nested {

 {

}

public class TestStatic {

 public static final double PI;
 static int counter = 10;

 static {

 out.println("1");

 }

 static {

 out.println("2");

 out.println("in inst init block");

 main() {

 out.println("3");

}

* * *
o/p *

- static initialiser "PI"
- another static block { all air memory allocated }
- main method
- non-static initialiser block.

? nonstatic initialiser

 main() {

 out.println("3");

}

Q45- Difference b/w instantiation block & constructor

5

3

public Box()

{

3

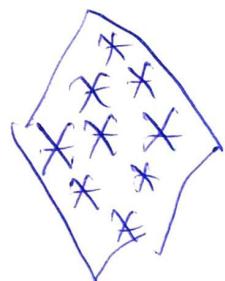
* Inheritance *

Why Inheritance :-

- (X) to avoid duplication by making inherit the class

~~(X)~~ Naming :- sub class child

Super class → parent



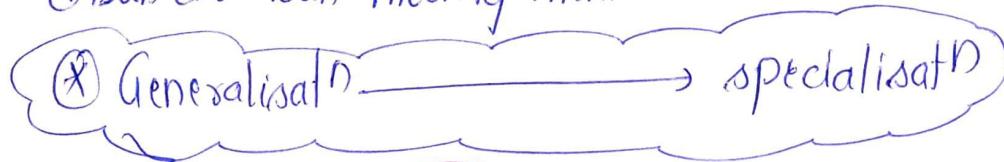
- ⊗ if we have common variable add them in "super"
 - ⊗ additional put int child
 - ⊗ Reusability is max

sub class:

- ④ Inherits all the member of parents & Grandparents..

- ④ house inherited

- ④ Sub class can modify methods.



- ④ if represent is a relationship

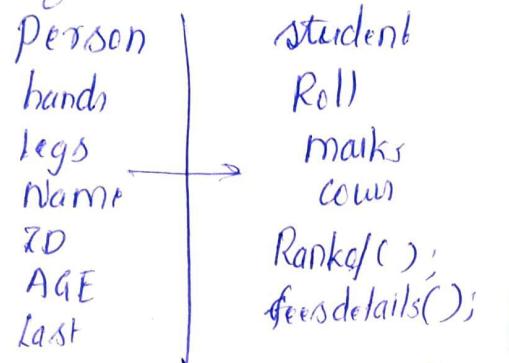
Key word :- extends.

Summary: sub class is a super class, and more (additional state + additional methods) and something modified (behaviour - method overriding)

④ Person, student, faculty

student is person

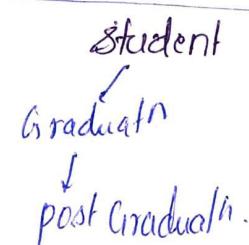
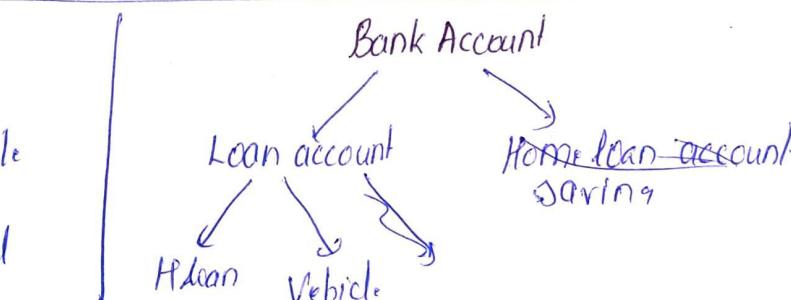
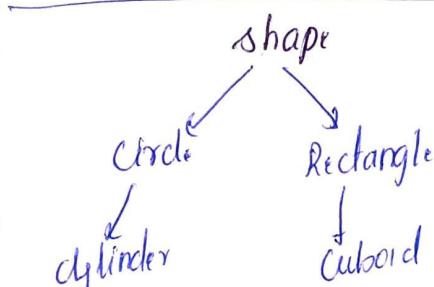
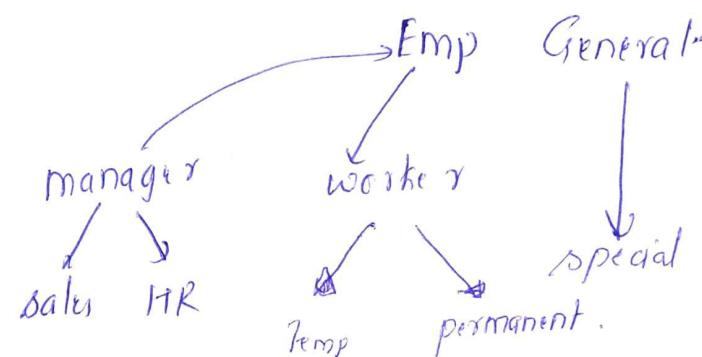
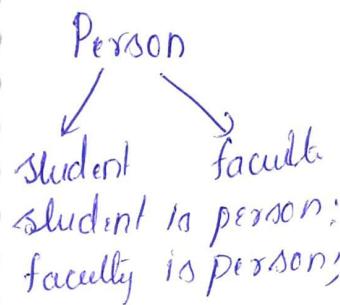
faculty is person.



if student inherits person

* Person, student, faculty

④ Emp, manager, sales mana, HR...



242 Types of inheritance

① single inheritance

class A { }

multilevel.

class A { }

{

class B extends A { }

{

class B extends A { }
implicitly multilevel
inheritance

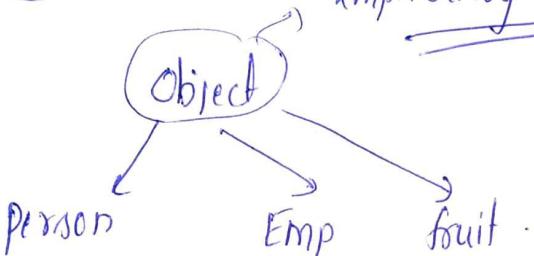
class C extends B { }

{

Implicit hierarchy

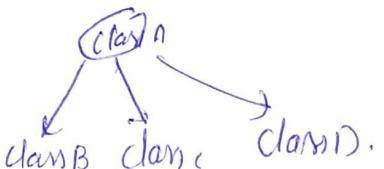
class A extends object { }

implicitly called



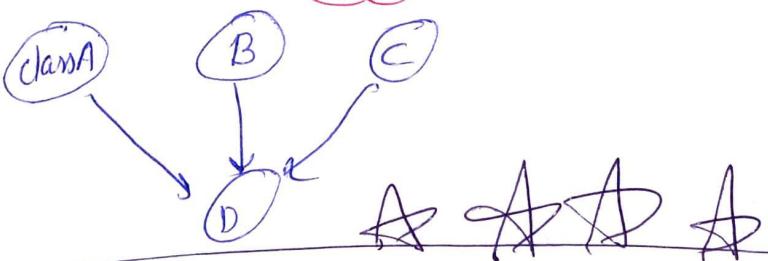
Hierarchical inheritance

when more than one class inherits from same class.



multiple inheritance

NOT supported → for simplicity.



Constructor invocation in inheritance hierarchy

*** Constructor Invocation *** Interview

Public class A {

A()

{
 System.out.println("A");
}

}

class B extends A {

B()

{
 System.out.println("B");
}

{

class C extends B {

C()

{
 System.out.println("C");
 // implicitly added super()
 // to get parents.
 // Meaning:- if invokes
 // immediate super class
 // constructor
}

}

Super

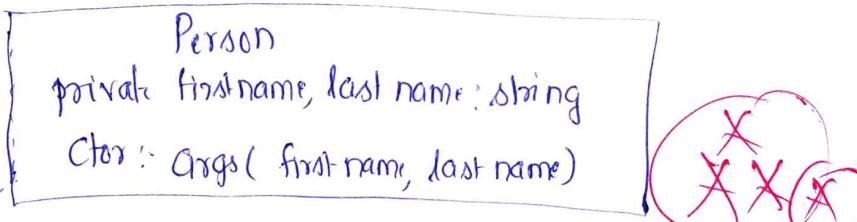
⊗ To invoke immediate super class constructor.

⊗ problem:- under tight encapsulation.

person: first name, last name,

student: first name, last name, grad year, course fees, marks

faculty: first name, last name, year of experience, S.M.T.



student

public class student
extends person { }

state: grad year, course fees, marks.

⊗ What happens if we don't write constructors

Compile time error

⊗ default constructor will be called.

⊗ super:- default will call default.

⊗ Solution: - To initialise complete state of student.

⊗ add parameterised constructor.

parametrised constructor for student

249 public Student (fn, ln, yr, co, fees, marks) {
 firstname = fn // Compile time error
 so let super class init the parent members.
 super(); → no matching in parent with
 X zero args.
 super(fn, ln); it will match with parametrised
 § Grad = Grad
 courses = courses
 fees = fees
 marks = marks
 }
④ sub class initialise its members and let parent initialise
 by super keyword.

10.21 11.6

Person

ctor: args -- firstname, lastname

public class faculty extends person

{ state yrs same
ctor: fn, ln, yr, smt
super(fn, ln);
}

Inheritance

Public class Person {

 private String firstName;

 private String lastName;

 public Person(String firstName, String lastName) {

 super();
 this.firstName = firstName;
 this.lastName = lastName;

 }

 System.out.println("In person constructor");

 Public () { } → parameterless constructor.

282

On new file

Public class Student extends Person {

 → implicit parameterless constructor added super()

 private int gradYear;

 private String courseName;

 private double fees;

 private int marks;

// add parameterized const.

 public Student(String fn, String ln, int gradYear, String courseName,
 double fees, int marks) {

// invoke parameterized matching super class ctor.

 super(fn, ln);

// init remaining members.

 this.gradYear = gradYear;

 this.courseName = courseName;

 this.fees = fees;

 this.int = fees;

 * first functn
 * in constructor

 * * * *

On new file

Public class Faculty extends Person {

 private int experienceYears;

 private String sname;

 public Faculty() { }

 super(fn, ln);

main() in different packages ~~Test~~

Public class Test1 {
 public static void main(String args) {
 Student s1 = new Student("a1", "b1", 2020, "DAC", "12345")
 }
}

* * first object class
person → loaded.
student →

~~(X)~~ In base class allot the private member also memory, they are derived from parent.

ctrl+mouse click → open declaratiⁿ.

faculty f₁ = new faculty (" " " " " ")

To display complete details of student & faculty (using overriding method).

class Person {

~~Always~~ Stick to one method and modifying method

class person{

```
public getDetails() {  
    //  
}
```

class student ↗ can modify

⑧ just inherit will get partial state.

Hence method overriding

Method overriding

1

Modifying the existing behaviour

* Person

student

faculty

adult
public getDetails()

public getDetails()

public getDetails()

Override.

* we have 3 copies of getDetails.

* Has java itself given you standard API for getting complete state? Yes.

To string()

**

*

go to object class → toString() returns string representation

public String toString()

↳ if not modified

if gets

get class().getName() + '@' + Integer.toHexString(hashCode())

String(hashCode())

fully qualified className

Tester main()

// print student details using inherited from person.

toString

sys0(s1.toString());

↳ gives methods

shortcut

sys0(s1)

Implicitly call s1.toString

HOW?

s1.toString

sys0(f1) (print faculty details)

O/P → inheritance student @ student H code.
inheritance faculty @

Usage of super

254

- ① Invoke superclass constructor.
- ② super.toString() any visible member.

mandatory else compiler will call this.toString()

~~faculty file~~

public String toString() {

return "faculty" + super.toString() + "Exp" + years + " ";

If we brief we comment overriding function is superclass.

④ toString in person.

⑤ will see hashcode and remaining details
instead of name from object class
& name.

Advantage of toString:

everywhere to get details.

Business Idea

write event organiser app this event student and faculty are invited.
prompt user for event capacity create suitable data structure to hold
participant detail?

option to user

① Register student

② Register faculty

③ Display participant details - for each.

④ Exit.

1st design :- suitable data structure → array of refs. of (inheritance)

person[] participants = new person[sc.nextInt()];

holder ↗ one object created.

④ Register student :

① Boundary checking

② Space.

Case 1:- `sop("Enter student details")`,
= create student type cast object
Student s = new student(...);

not adding into array.

Participant → problem → one side person on other student.

Participant [counter++] = new student();;

No error.

Student is person,

inheritance.

↑ climbing up the hierarchy

implicit conversion.

** UPCASTING

Compiler implements inbuilt type cast.

Person p = new student(); → Upcasting
climbing up the hierarchy.

main() {

// super class ref, super class obj, sub class ref, sub class

// Person : super class, student, faculty : sub class

direct and
indirect reference Person p; super class ref
System.out.println(p), uninitialised hence error

P = new person("a", "b"); → Super class object.
super class

? Student s1 = new student("a", "b", 2020, PDATA);

sub class ref → Sub class object;

? P = s1; → No error

S1 is a P

- 4:05 11 In run time : $p \rightarrow$ pointing to subclass object; 256
~~System.out.println(p);~~ → P. casting implicitly.
- ⊗ P → Java compiler resolve call by the type of ref.
 - ⊗ JVM resolves by Type of object referred by reference.
 - ⊗ Casting is person → student → automatically called
faculty →
 - ⊗ Java this decision is taken by JVM: Runtime Polymorphism
 NO virtual keyword → all methods are virtual

$p = \text{new faculty}()$ → Compiler checks faculty IS A Person.
 $\text{System.out.println}(p);$ → Java compiler goes by type of ref;
 JVM goes by object type.

 ⊗ Decision of calling toString during run time is JVM called as runtime polymorphism;

Event Organiser: main() {

 Scanner sc = new Scanner();

 System.out.print("Enter event capacity");

 int capacity = sc.nextInt();

 Person[] participants = new Person[capacity];

 int counter = 0;

 boolean exit = false;

 while(!exit) {

 System.out.println("1. Student 2. Faculty 3. View 4. Exit");

 System.out.print("Choose");

 switch(sc.nextInt()) {

 case 1: if(!student[counter].checkBoundary())

 if(counter < participant.length) {

 System.out.println("Student");

 Person p = new Student(sc.nextInt());

 participants[counter] = p;

 counter++;

 }

 else {

 System.out.println("Event full");

 }

 }

 }

 }

25F case 2 : if (counter < participant.length) {
 sys("Enter faculty detail")
 participant[counter++] = facultyNewFaculty (sc.nextLine());

case 3 : //display

for (participant)

 person p; participant[i] is

 if (p != null) {

 sys(p);



Run time poly morphism → automatically checked by JVM runtime.

To avoid nulls checks



Q Interview → How to initialise complete state of subclass?

* super(fn, nm);

① How many object created when subclass creates object?

⊗ only 1 lab

16:31

⊗

Inherit to string and modify

To access getters of super class.

super.getFirstName.
it is optional

⊗ Direct reference

same ref same obj

⊗

