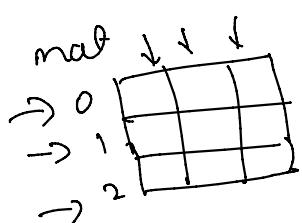


## Arrays in Java contd...

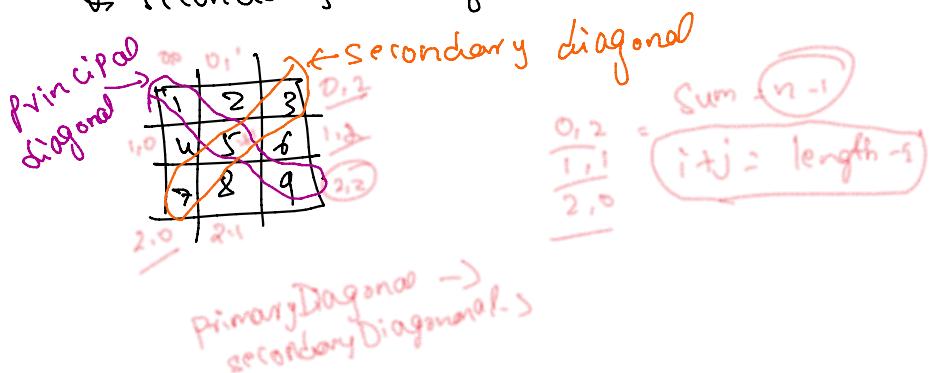
\* Read & print a <sup>in</sup> matrix

```
int [3] array = new int [3];
int [ ] [ ] mat = new int [3] [3];
```

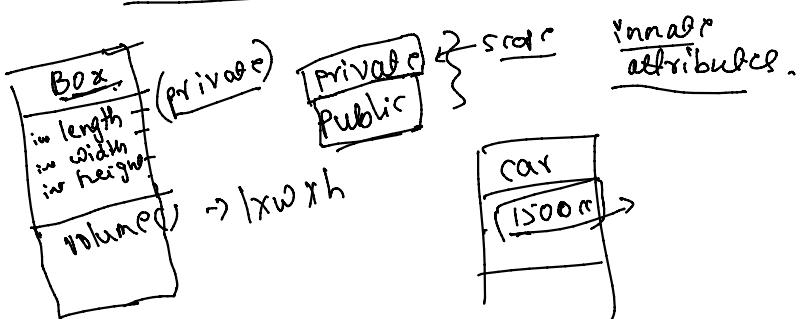


mat.length  
mat[0].

\* Read a matrix, print principal & secondary diagonal elements.



: Classes & Objects :



```
class Box
{
    private int length;
    private int width;
    private int height;
}
```

```

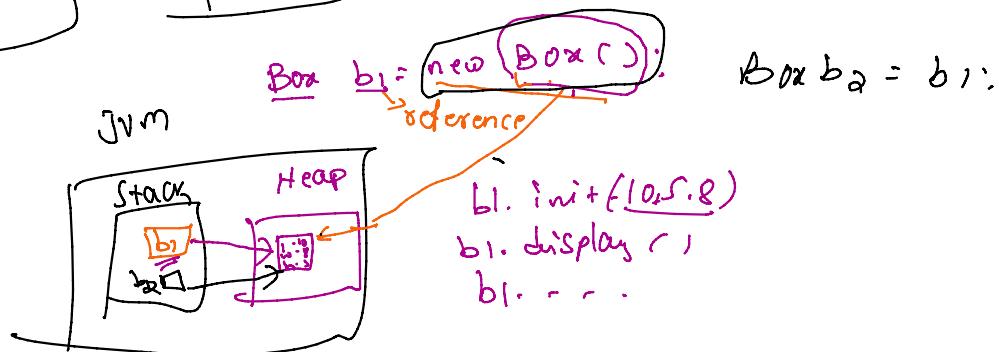
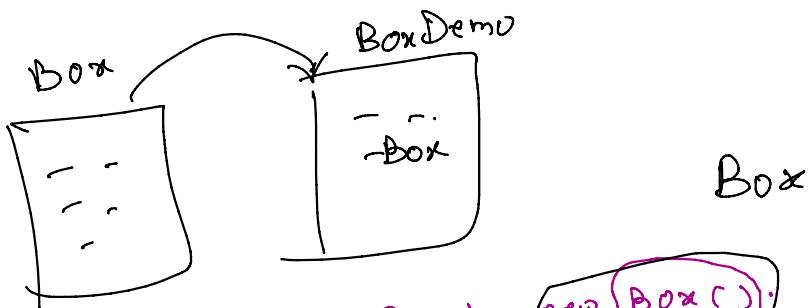
private
private int height;
public void initialize()

```

```

public display - l,w,h
public volume = l*w*h

```



+ Create a class Laptop e.g

attributes: RAM - e.g  
int HDD -

private {  
String processor -  
double screenSize -  
int intel/AMD  
15.2  
13.2  
17.2  
}

function: methods: initialize()  
display() → all features.

Laptop hp = new Laptop();  
Laptop dell = new Laptop();  
int init dell & disp;

init  $\rightarrow$

constructor: method / function.

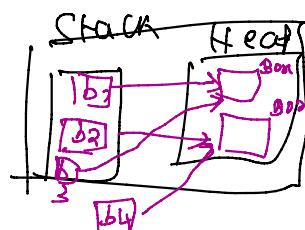
- \* name = name of class  $\xrightarrow{\text{public}}$
- \* No return type (Not even void)
- \* called when an object is created.  
gets called once

Box b1 = new Box();

Box b2 = new Box();

Box b3 = b1;

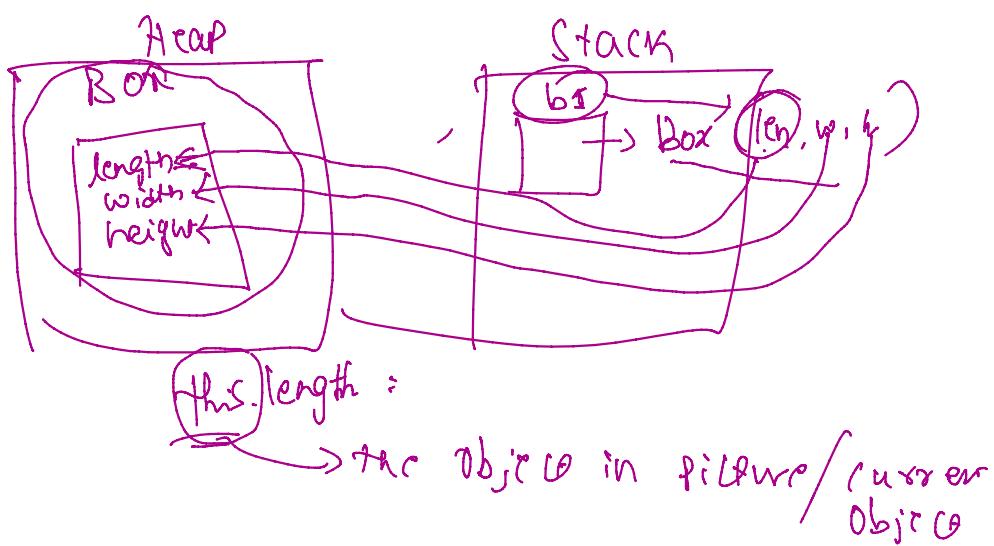
Box b4 = b2;



Pars parameters to constructor.

W/o parameter  $\rightarrow$  default constr.

With ()  $\rightarrow$  Parameterized constr.



... getters & Setters for

: getters & setters for  
the attributes:

Box → l = 10, w = 20, h = 30  
int

public int getLength()  
g  
return length;

→  
public void setLength(int length)  
g  
this.length = length;

→

\* Create a Box Object. with values

l = 100

w = 50

h = 200

→ find the volume. →

→ increase the (length by 10 %), width by  
20%, height by 8%.

→ display l, b, h

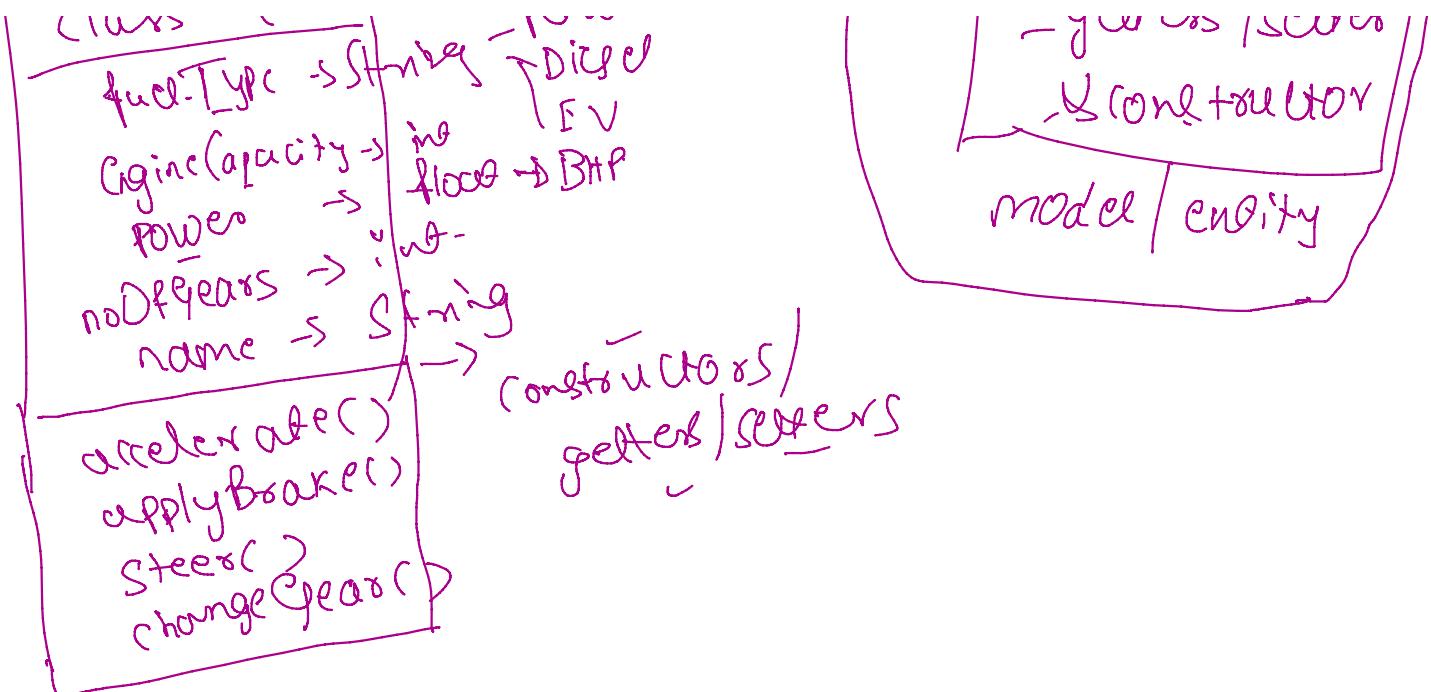
→ display volume.

POJO → Plain Old  
Java Object

Object  
attributes  
- getters / setters  
constructor

Class Car

fuelType → String  
person → Person



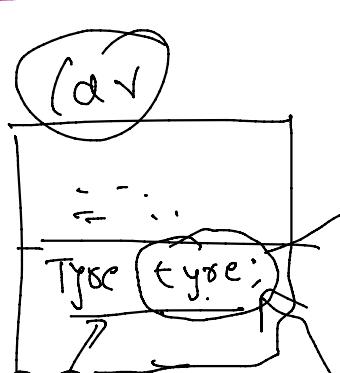
\* Create a more cars (of your choice)  
Rolls Royce, Audi  
display data., drive the car

→ changeGear method, modify it so that  
it accepts (gear No.)



⑥ ⑤ + ①

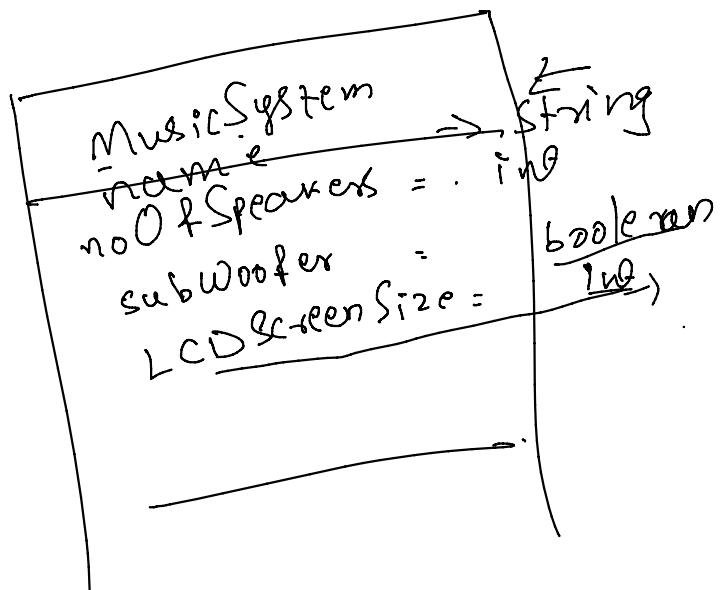
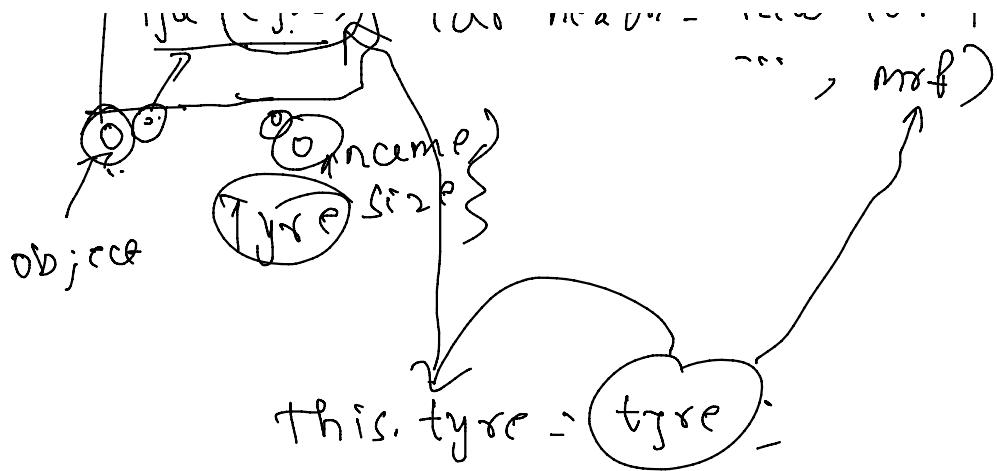
Type . type:



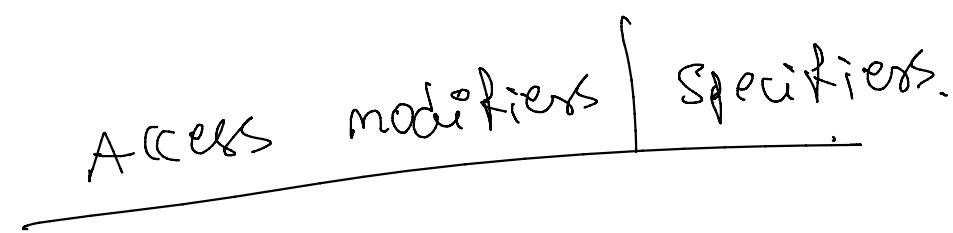
Tyre (mrf)

new Tyre("MRF", 10);

(or nexton = new Car ("Nexon" . . . , mrf))



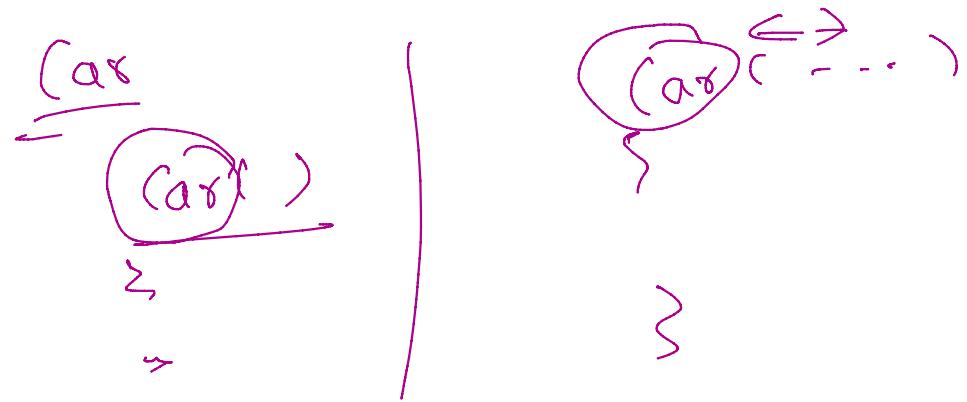
- \* Create MultiSystem class, & add it to car & display the details
- \* Change MusicSystem later.



private  
public  
protected  
default  
(package)

(Method  $\uparrow$  overloading:  
dis, by (float f))

void display(→)  
display(← a)



displayFeatures ( - - )

{      \*

-

displayFeatures

\* if (String = &tyre) →

>

