

# Object Oriented Programming

binary  
assembly also

manipulate objects

t=10  
m=7

b=9

print(t+b)

print(t\*m\*b)

unstructured not well sequence of instruction

vector  
.step  
high level language  
↓  
defined Goto - for repetition

structured entry point

if

=

else

for

=

while

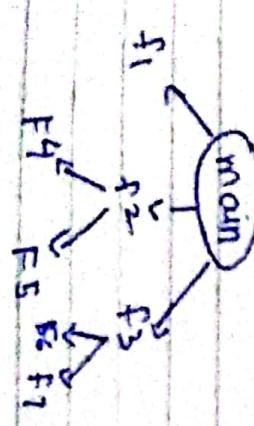
=

do

=

end

=



objects:  
invokes function to process them

object has data and functionality



in procedural programming:

teach("classes", "oop", "BT")

In OOP: (bottom-up approach) change break  
ideas. teach("OOP", "BT") has had how,  
object banana, pose

→ every object is independent

→ code manageable  
→ technique of writing better code

15/6/23

### functions

- a) built in
- b) user-defined

types:

- a) built in      on broader terms

- b) user-defined

def func\_name:

class class\_name:

→ definition of class

class Vector:

pose



def    =

def   :

→ a type now

class   :  
pose

a=10

a → int

10

corresponding

a=10

b=20

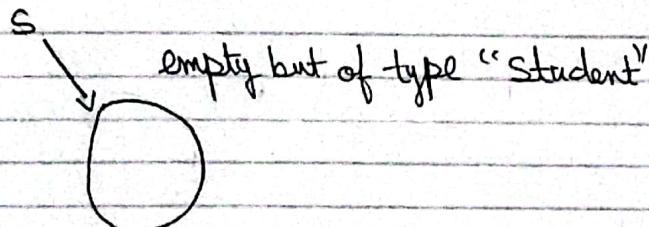
v1 → vector

v2 = vector()  
v2 → vector



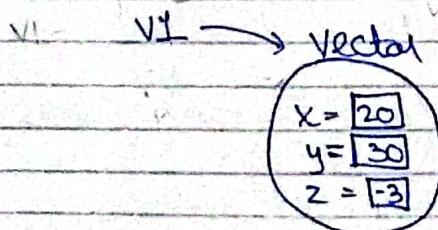
class Student:  
    pass

s = student()



→ int is a type  
→ int() is also a function

v1.x = 20  
v1.y = 30  
v1.z = -3



print(v1.y)  
→ 30

v2 → Vector

x → 30  
z → 90  
y = 2

rollno = "BScs...."

v2. x = 30

v2. y = 90

v2. z = 2

component of object  
| data  
members of v2

v2.rollno = "BScs...."

class pehley baratay  
then object baratay

identifier: function, variables, class ka name  
are called identifiers.

CPU ka time kam nhii hota

s = stp(t, m, b):

↓  
better because wahaq parameters  
hainay chahiye

b.z = -2

b.institute = 88

b.z = 88

↓  
diff

naya ban jaayga

agai exist karta hai and value lagaein  
tak update

```
s = Student()  
s.rollno = 'BSPS...'...  
s.name = "Arshad..."  
s.cgpa = 3.17
```

Student

rollno = BS-  
name = 'Arshad'  
s.address = '...'

Address()

is a class of this  
type is available so  
it is okay

s.address.city = 'Lahore'

int + int

int / int

\*

powel (int1, int2)

st (input())

no functionality is available with python vector,  
no add, no sum, no divide

if v2 is vector():

print(v2)

⇒ error

v=vector(3,2,1)

int + b

t+b

vectors ka bhi add hona chahiye

list + list

def print\_vector(v):

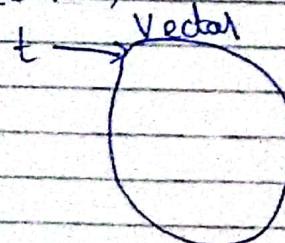
ziyada ter karein gaayi aagayi seh.

class

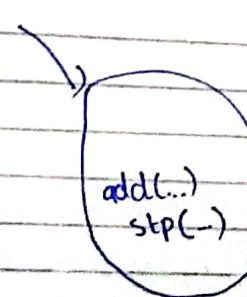
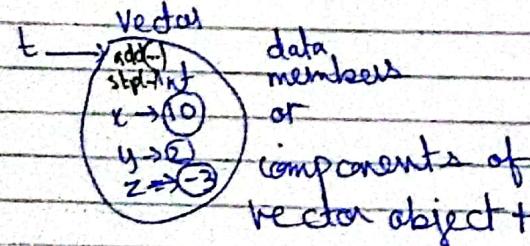
• operator

functionality khud banani hoti hai

t = Vector()



t.v = 10



t.add = existing  
function name X not appropriate  
define at class level

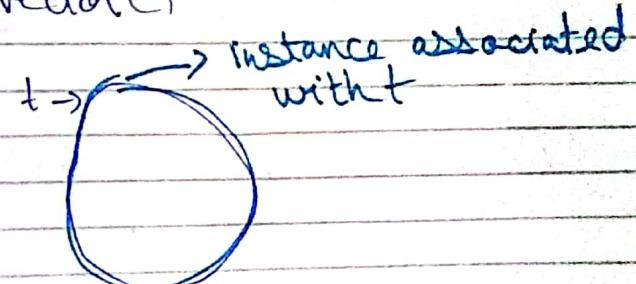
t.v = ?  
print(t.v)

t.v + 2

t.add  
func type:

- 1) Instance members (value of a class)
- 2) static members
- 3) class-level members

t = vector()



$t+b$   
 $t.add(b)$   
 $t.x+b$   
 $t.slp(t,m,b)$   
 $t.slp(m,b)$   
 caller object

$t.reverse()$

$x = -10$   
 $y = -2$   
 $z = +3$

32-6-23

$t=Vector.add()$

static method (slp)

$t.add(b) \rightarrow$  PF leave

$t.add(b)$  // instance

$Vector.add(t,b)$  static // best

$Vector.reverse(t)$   
 $t.reverse()$

automatically called

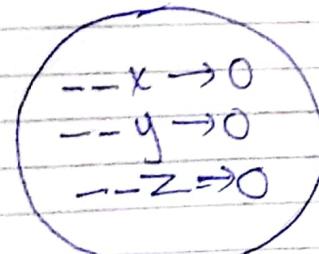
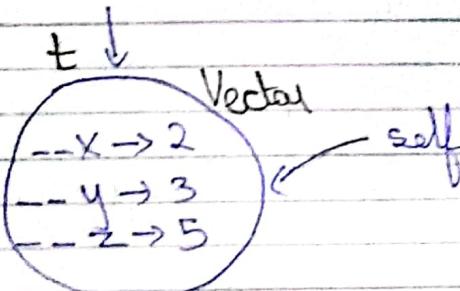
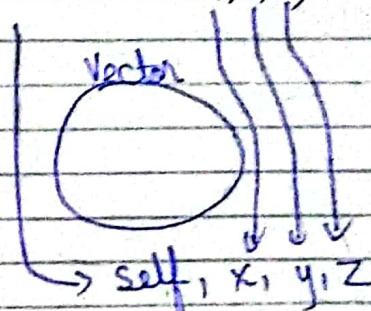
$\text{def } \text{init}(\text{self}, x=0, y=0, z=0):$   
 $\text{self}.x=x$   
 $\text{self}.y=y$   
 $\text{self}.z=z$

Non  
 static aayega.  
 system calls  
 hooga

10, 10, 10 bhi  
 ho sakte,  
 but kisi  
 seh initialization  
 ho sakte  
 hai

① ② ③ ④ → parameters

$t=Vector(2,3,5)$



X inappropriate

t. \_\_init\_\_(1000, 1001, 1002)

explicitly call init

init function automatically aik data call  
hota hai

static bhi available hai

def \_\_str\_\_(self):

```
vstr += "("  
vstr += str(self) + "-x)  
vstr += ", "  
vstr += str(self) + "-y)  
vstr += ", "  
i
```

str ka aik hi parameter hota hai

scope of a variable

a = 10  
def abc(x):  
 x = 90  
 print(x)

abc(a)  
print(a)

output:

90  
10

main function  
scope

i: 10

abc

func abc  
scope  
return

↓  
after it everything is  
deleted

def abc(x):  
 v.m

A diff break def → reserved words

a → variable

Ixyz XX not an identifier.

class ✓  
pass → user defined data types  
structures

Enumerators - list mein choti si aik value

DS  
CS  
IT  
SE → small list

s = DS  
s = "DS" but program of a  
student is not a string

class provinceOf Pak:

Sindh = 0

Kpk = 1

Punjab = 2

Balochistan = 3

P = province Of Pak. Sindh

enumerators seh  
codes khoobsurat

formal parameters → arguments

def xyz(a,b):

actual parameters

xyz(p,q)

python mein self call by reference hota hai

ctypes arrays, numpy arrays, binary file handling  
local variable, global variable, scope,  
Lifetime, exception handling, first-class  
objects, lambda, call back shallow vs  
deep copy, mutable vs immutable types, tuples,  
set, dictionaries, aliasing, instance member  
raise → exception handling function

try/catch

Lambda

first class object → functions vohi

class Vector:

pass

k = Vector

O = k()

O.k =

O.y =

O.z =

abcvector,

def abc(...)

f = abc

f(...)

`x = [2, 3, 7]`

`y = (2, 3, 7)` → tuple (read only)

`z = ( )`

`y[0] = 7` X  
`y.append(a)` X

`t = y + z` (-----, -----)  
`y = y + (2, 3)`

`z = {2, 3, 7}`

set  
not indexable  
mutable

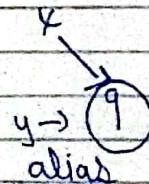
dictionary &

Lahore: 43  
Karachi: 45

aliasing

`v = 9`

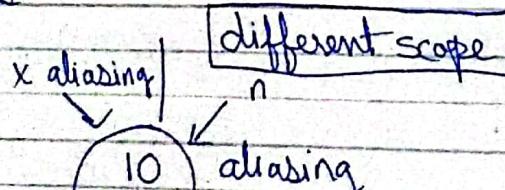
`y = x`



`x = 10`

`def abc(n):`  
=====

`abc(x)`



ctype arrays

numpy arrays

binary file handling

instance member function

`def fn(self, ...)`

zaroorat nahi self likhna  
keh liye

init function

initialise karney mein use  
hota

- class mein hogा tаh  
atleast aik data call hogi

`str`

`a = [1, 2]`  
`b = a`

static method ] - use either `b = a[:]`  
`t.`

`b = b[:: -1]`

`v = vector()`

`v = v.reverse`

getters, setters, decorators

`@staticmethod`

decorators hamisha deh array start hota

## Pillars of OOP:

encapsulation : data and code are combined  
as a unit (class).

Inheritance

polymorphism

abstraction

int(x)

str(...)

x

y

z

add()

details

↳ information hiding (abstraction)

— x → data member

V17



— x → private  
member not accessible in main()

private  
&  
public

members of objects

all members of class `private` name \_\_ (a underscore) eet bura hoga woh private mark  
hoga, can't be accessible outside class

private data ka function bhi through access  
Karakter

`print(m.qtyY())`

faida?

- when you are using

rational number

p, q ≠ 0

q

rational()

r, q ≠ 0

r, p ≠ 0

r, — q ≠ 0 X

`def setQ(self, q)`

if q != 0:

self.q = q

else:

=

getters / setters:

protects the data so it can't be invalid  
at any stage.

`self.setX(x)`

`def setPwre(self, p):`  
`self._pwre = 0`

and use ko koi  
prepective dikha  
rahey

s.setRollNo("BSPSF22M079")  
slicing kar ke save karay

prog: "BSNS"  
sess: "F22"  
section: "M"  
rollno(sn): 79

end user → dumb

def diff(a,b):  
 return a-b

x=10

y=3

print(diff(x,y))

x → 10

y → 3

output : 7

now,

if print(diff(x)) ~~xx error~~

def diff(a=7, b=3):

print(diff(x)) ✓  
 ↑  
 default values

## Privatisation of data members

### data members

→ getters ✓ setters ✓ normal

→ getters ✓ setters ✗ → read only, write  
 nh

→ getters ✗ setters ✓ → write only, not  
 readable

→ ✗ ✗ → unacceptable to pull  
 publicly inaccessible

### magic members (functions)

fun-name \_\_

init \_\_

str \_\_

add \_\_

+ b → isko likhna ch

t... add... (b) ←

get width call  
data

t... eq... (b)

↑ ↑

t=b call wohi hojata, yeh likhna ch

abs\_()

↑  
abs(v)

abs → magnitude of its  
function

v[0]

v[1]

v[2]

# getters/setters & decorators

class Vector:

def \_\_init\_\_(self, x=0, y=0, z=0):

self.setX(x)

self.setY(y)

self.setZ(z)

def setX(self, d):

self.x = d

return

def getX(self):

return self.x

def setY(self, d):

self.y = d

def getY(self):

return self.y

def setZ(self, d):

self.z = d

def getZ(self):

return self.z

def str\_(self):

return "(" + str(self.getX()) + ", " +  
str(self.getY()) + ", " + str(self.getZ())  
+ ")")

def \_\_add\_\_(lhs, rhs):

v = Vector()

v.setX(lhs.getX() + rhs.getX())

v.setY(lhs.getY() + rhs.getY())

v.setZ(lhs.getZ() + rhs.getZ())

return v

@staticmethod

def dot(v1, v2, v3):

v = v1.getX() \* (v2.getY() \* v3.getZ())

- v1.getZ() \* v3.getY())

v = v1.getY() \* (v2.getX() \* v3.getZ())

- v2.getZ() \* v3.getY())

v += v1.getZ() \* (v2.getX() \* v3.getY())

- v2.getY() \* v3.getX())

return v

def main():

t = Vector(2, 1, 5)

m = Vector(3, -2, 4)

b = Vector(4, -1, 2)

print("t: " + str(t))

print("m: " + str(m))

print("b: " + str(b))

print("t+b: " + str(t+b))

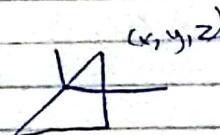
print("dot: " + str(Vector.dot(t, m, b)))

main()

21/9/23

self → is not a keyword

↳ just a widely used word



(x, y, z)  
Write a program to  
2 vectors input karey  
add karey

print resultant vector

$$v_1 = x_1 \hat{i} + y_1 \hat{j} + z_1 \hat{k}$$

$$v_2 = x_2 \hat{i} + y_2 \hat{j} + z_2 \hat{k}$$

$$\text{ans} = x_1 \hat{i} + y_1 \hat{j} + z_1 \hat{k}$$

coord.

x1 = int(input("Enter value of i-coord. of vector 1: "))

y1 = int(input("Enter value of j-coord. of vector 1: "))

z1 = int(input("Enter value of k-coord.  
of vector 1: "))

x2 = int(input("Enter value of i-coord.  
of vector 2: "))

y2 = int(input("Enter value of j-coord.  
of vector 2: "))

```
i1 = int(input("Enter 1st integer: "))  
i2 = int(input("Enter 2nd integer: "))  
i3 = i1 + i2  
print("Their sum is", i3)
```

Again try to take both float nos.

```
v1 = vector(input("Enter a vector"))
```

```
v2 = vector(input("Enter another vector"))
```

i3 = v1 + v2

print("Their sum is", i3)

step

|     |     |     |
|-----|-----|-----|
| (x) | (y) | (z) |
| mx  | my  | mz  |
| bx  | by  | bz  |

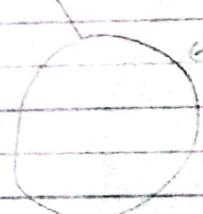
```
c = complex(2, 3) → 2 + 3j b/w type & type  
print(c)
```

complex → built-in  
type

Assignment : Tuesday

v1 = Vector() # creates empty object

v1



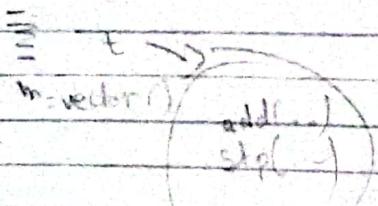
ekhans hoga  
jst class mn hichat

v1.x = 3      is v1.x a valid name?  
no

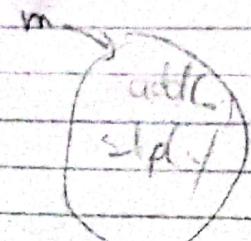
if main expression is  
the opposite sign of

i=10      → if i exists toh  
update ho mtlha waha  
aata hoga

t = Vector()



m.add



$$a+b = b+a$$

whilr add matters, std is arbitrary

$$\frac{2}{3} + \frac{4}{5}$$

rational / fraction / ratio

make a class for rational fractions

num

class rational:

def \_\_init\_\_(self):  
    r1 = rational()

num

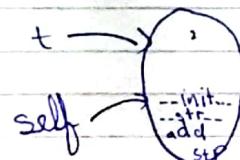
3 7

make atleast 2  
functions

- (1) multiply( $m$ )
- (2) real( $m$ )

add      0.3      3  
              10      20  
subtract  
reciprocal

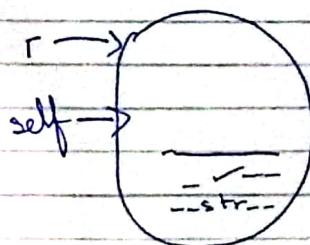
caller object



a = None

↳ keyword for nothing

a = 10



x=1, y=2, z=3

r=Vector(x, y, z) ??

↳ Vector (v1.x, v2.y, v1.x+v2.y)

print(f "t: {t}") // string called here

t.add(b)

↑

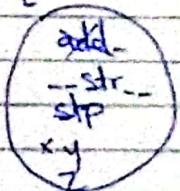
# instance member function

# static method , static member function  
calling a Vector.add(t, b)

Static method se h

Vector main add  
member  
hata har

+ ka member  
function nahi hota  
phir



Syntax:

def add(v1, v2):

static method (add)

[init, str] cannot be made static method

↳ can write it anywhere, no need to write it immediately, it is a function

Private data members:

self. → x

can't print directly in main

print(t.\_\_x) // error (-x is not defined)

→ inside the class accessible

→ not accessible outside the class

c = complex(2,5)

c.r = 10

↳ if not available will be created,  
otherwise updated

@staticmethod  
def str(...):

→ decorators  
/ annotations

function becomes defined as  
a static function

sd mat lab oap varen aap boundary conditions  
and other conditions ko check karey

time span  
months

to timedelta

L.Hours = 7  
x.t.Hours

④ property → decorated  
def Hours(self):  
 return self.hrs

④ Hours.setter

def Hours(self, d):  
 self.hrs = d

main file work kar jaa rha hua karey hain to  
if \_\_name\_\_ == "\_\_main\_\_":

main()

agar main ray file export ki tab woh main  
func nahi call karey

Tu main yahan main hota wo hoga  
koi jaha doctori file ki main ka nahi call  
Karey qf upr watare karey qf

constructor initialiser, tk

date

dt / dt/yy

range checks bhi lagayegi parhar  
gaya

time

hh:mm:ss/f

complex numbers

fractions

## Polynomials

$$(p_1: 2x^3 - x^2 + 5)$$

$$(p_2: 3x^2 + 2x - 7)$$

$$2x^6 + 3x^2 + 2x - 2$$

can be divided / added  
with playing

$$\text{at } x=10:$$

$$\text{ord}(\cdot)$$

p = Polynomial(...)

Identify data members filled

- var = 'x'  
- deg = 3  
- coef = [3, 0, 0, 1]

- var = 'x'  
- deg = 2  
- coef = [3, 2, 1]

{3, 0, 0, 1, 2, 1}

3 2 1 0  
index from back to front

@property

def val(self):  
return l\_val

(can put conditions  
to since it's a func)

call

properties

\_\_init\_\_ -- add --

def \_\_init\_\_(self, v, d, c):  
var=v

p = Polynomial('x', 4, {2, 0, 0, 1, 3})

ignore 0s

$$2x^3 - 3x^2 + 4x + 0$$

how: make polynomial class

constructor:  $\text{__init__}$

Destructor

- cleanup code to free resources
- resource statistics etc.
- destructor → control the creation of objects

Final destructor:



class Vector:

=

c.Vector



allowed in Python



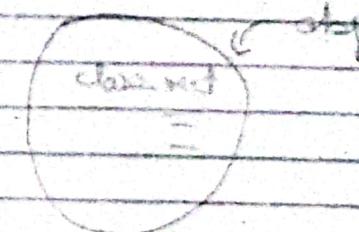
def \_\_del\_\_(self, ...):

del f()

f = None

class Vector:

```
def __new__(cls, x=0, y=0, z=0):  
    obj = super().__new__(cls)  
    Mdc()  
    return obj
```



SINGELTON:

class Vector:

```
def __new__(cls, x=0, y=0, z=0):  
    obj = cls()
```

# infinite loop

## function overloading

functions with same name in a scope

```
def sum(a,b):  
    ==
```

```
def sum(a,b,c):  
    ==
```

it is not allowed in python

diff languages mean:

through different count and orientation of parameters

```
def sum(a,b):  
    return a+b
```

sum(12,1)  
sum(12.5, 11.5)

sum(12, 11.5)

sum(12.235967, 0.00001)

```
print(sum(2,5))
```

```
x = sum(2.1, 3.2)
```

```
print(sum(complex(2,3), complex(3,7)))
```

```
print(sum(Vedant(2,2), Vedant(3,3)))
```

```
def sum(a,b):  
    return a+b
```

a=None, b=None, c=None  
def sum(a,b):

if type(a) == type(1)

if

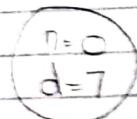
{variable no. of argument  
keyword argument}

function overloading is not directly available

student obj class roll no.  
Phone num id no. mother name

STATE of an object :

values (objects) associated with data members of objects



$$\text{state} \rightarrow \frac{10}{7}$$

It should remain valid ~~all the time~~ for whole life of object.



$$\frac{10}{0} \rightarrow \text{not a real number}$$

ratio (2.5)

$\frac{2}{5}$

factor (7)

7

ratio ()

0

class Bill:

Total  $\rightarrow$  int

Address  $\rightarrow$  string

Name  $\rightarrow$  string

No  $\rightarrow$  int

B = Bill( ... )  
print(b)

Total  $\rightarrow$  int

Amount  $\rightarrow$  int

particulars  $\rightarrow$  string [list of string]

qty  $\rightarrow$  int (list) of int

Rate  $\rightarrow$  int string

Date  $\rightarrow$  string (date)

Name str or can be of type name

Address  $\rightarrow$  str (Address)

Rate  $\rightarrow$  int (array of int)

Signature  $\rightarrow$  string

Alt  $\rightarrow$  string

diff in time (to get age)

Qty  
Rate  
Particulars

— b1[0].qty  
b1[2].Rate  
b1[1].particulars  
array of obj. etc

class BillLine:

BL = 23

bl.append(BillLine(2, 6.5))

rate[]

members

↓

name → int ✓

date → date?

name → string

address → address?

qfg -

pt -

rate

class BillLine

Instance <sup>local</sup> members vs class <sup>local</sup> members

--glg:D

shared  
among all  
obj of class

update = file (20/04/2022)

update = file (20/04/2022)

update = file (20/04/2022)

→ 20/04/2022

int max (int Sabit, int end, string  
member)

string split()

```

def average(a, b, c):
    {
        return avg
    }

```

→ definition of function

describe: exact seh thara sa kam

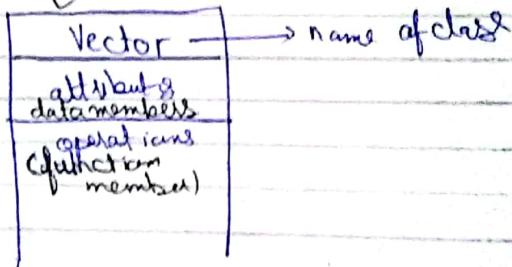
teendas can come in paper

print("this func kills returns...")

spontaneous (not predictable)

UML :- unified modeling language

class diagram, activity diagram, etc



name of class

C stands for class

### C Vector

negative

j is key  
s a nth

with private  
member

positive

j is key s a nth

with public  
member

-x: float

-y: float

-z: float

+ init(x: float, y: float, z: float)

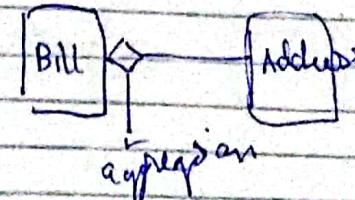
+ add(v: vector): vector

UML is not for python

Composition & Aggregation

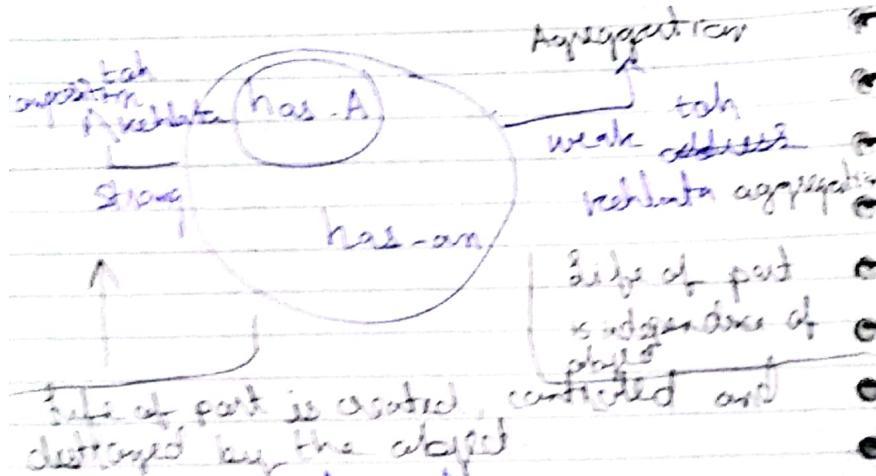


composition

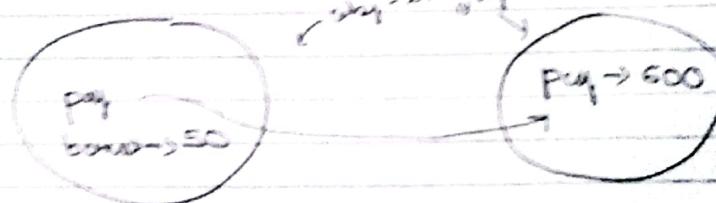


aggregation

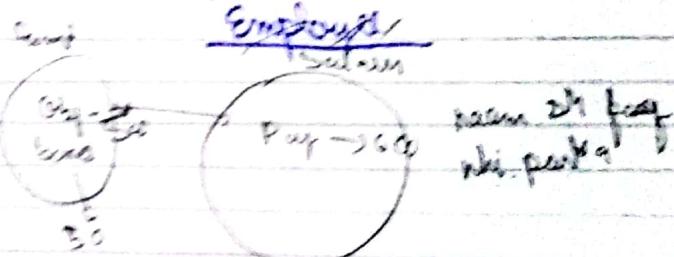
A bill has an address  
A bill has items on it



Aggregation



Cong. Total 770.



In composition tree is only 1 variable

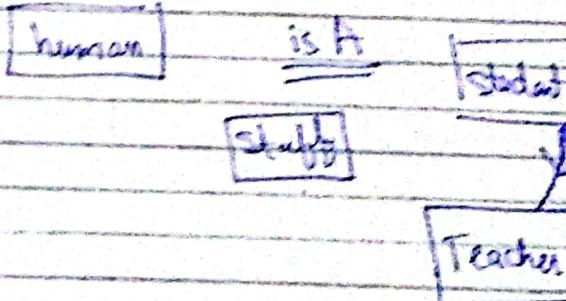
part of many objects but has their own  
identity.

Composition many many create Relation

Q

Centres of Circles only exists with its circle.

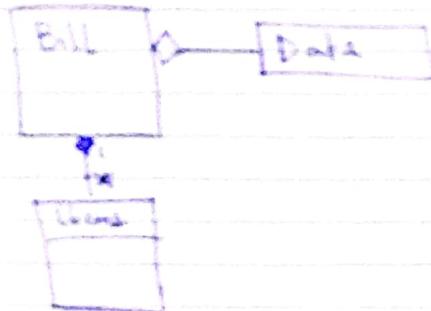
classes



Teacher is a human  
not every human is teacher

Inheritance heritage ▷

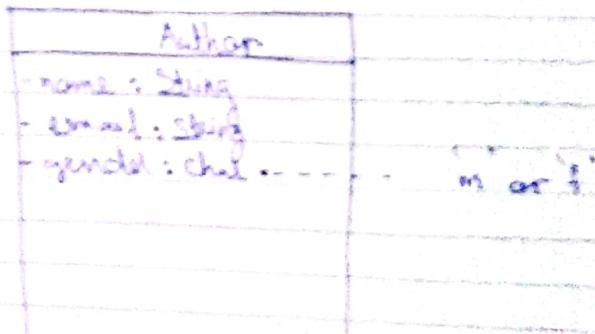
Composition - instance (can't destroy by going out)



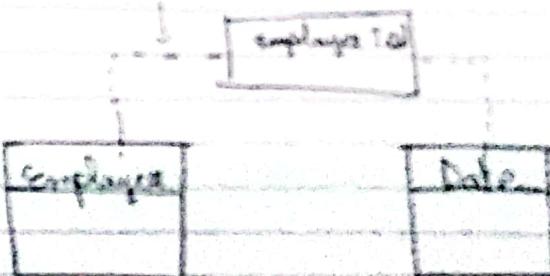
Aggregation - has-a

Composition - Part-of

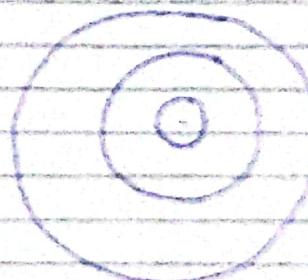
Association - has - a



Redirection



Circle point



Concentric circles