


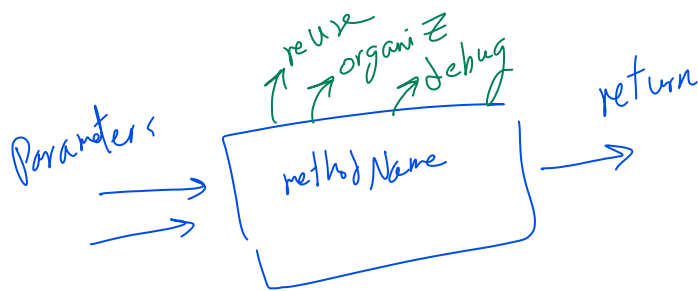
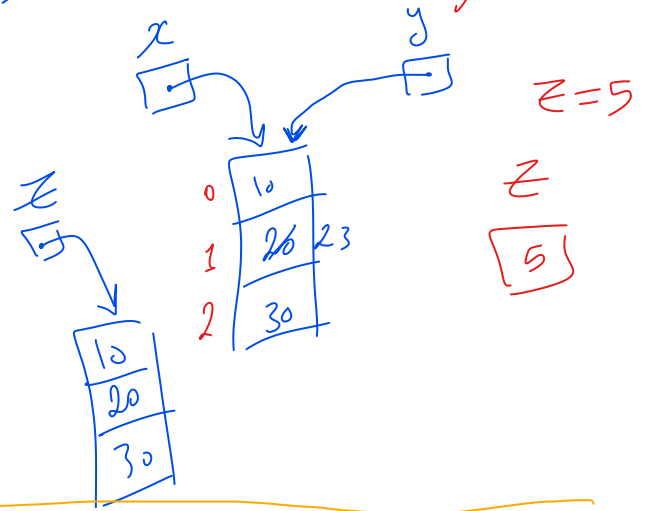
- * Arrays
- * Methods
- * Classes & Objects

```
int[] x = {10, 20, 30};
int[] y = x;
x[1] += 3;
```

Arrays.



Reference Datatype



Call Method

`Math.random()`
`Array.copyAt(-1)`
`new.charAt(0) → 0`

modifiers return-type methodName (datatype pName, ...) {

}

weight height



calcBMI(---)

$$bmi = \frac{weight}{(height / 100)^2}$$

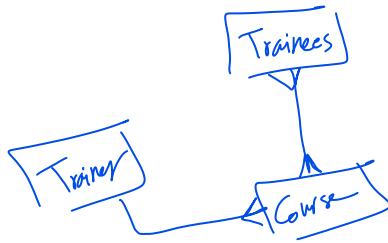
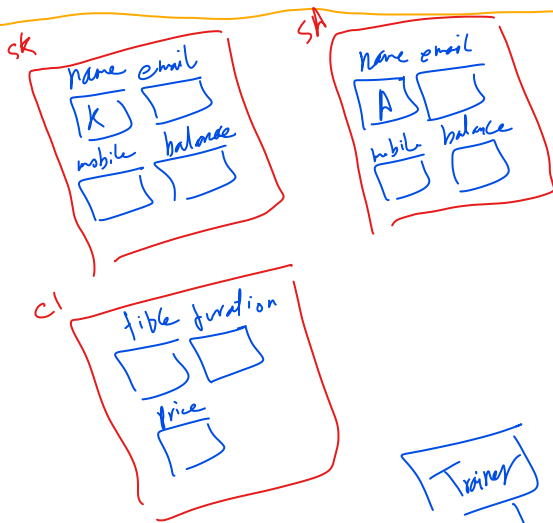
getStatus(bmi)

< 18 Underweight
 < 25 Normal
 < 30 Overweight
 Obese

mark FullMark

$$Pct = \frac{\text{mark}}{\text{FullMark}} \times 100$$

>85 Excellent
 >75 V. Good
 >65 Good
 >=50 Pass
 Fail



OOP

- object \leftrightarrow special variable place memory
- class \rightarrow type
 - code
 - create & init object
 - define operations

name weight height

--	--	--

$$BMI = \frac{\text{weight}}{(\text{height}/100)^2}$$

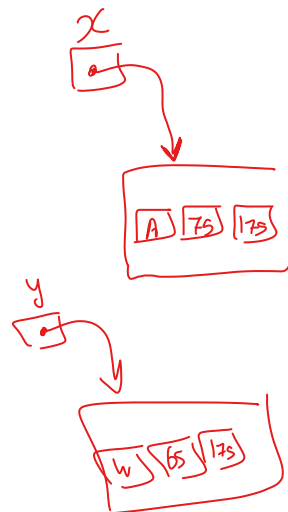
OOP

Patient

name
weight
height

Patient (-1-1-)

calcBMI()
getStatus()





$$\text{annualSalary} = \text{salary} \times 12$$

$$\text{servicePeriod} = 2024 - \text{hireYear}$$

OOP

Employee

name
salary
hireYear

Employee(-, -, -)
calcAS()
calcSP()



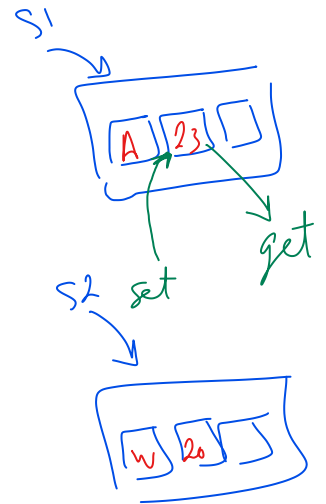
$$\text{pct} = \frac{\text{mark}}{\text{fullMark}} \times 100$$

OOP

Student

instance variables { name
mark
fullMark
static variable ←

Student(-, -, -)
getPercent()
getGrade()

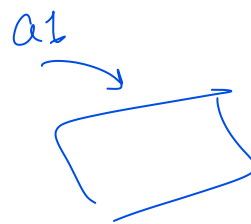


Encapsulation

Account

name
- balance
annualRate = 0.04

Account(-, -)
getMonthlyRate() → / 12
getMonthlyInt() → balance *
deposit(amt)
withdraw(amt)
getBalance()



a1.getMonthRate()
a2.getMonthRate()

