Today's agenda

- float and double
- Operators
- if else

// Decimal → 2.2, 1.3, 10.0, 3.76 etc.

↳ float & double

float → int → (.6-7 decimal places)

double → long → (.14-15 decimal places)

→ In integer, default values are int.

→ In decimal, default values are double.

Integer

↳ long l = 10^10L;

Decimal

↳ float f = 2.4f; → write f at the end while initializing float.

* 2 Golden rules of typecasting

↳1. if there is guaranted no loss of data: implicit

Ex: int to long.

float to double.

↳2. if there is a chance for loss of data then:

we can still do the conversion forcefully: explicit

Ex: long to int

double to float

Quiz 1:

```
double d = 2.8;
System.out.println(d);
```

→ 2.8

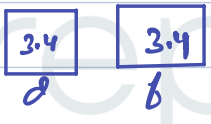
Quiz 2:

```
float f = 3.3f;
System.out.println(f);
```

→ 3.3

Quiz 3:

```
float f = 3.4f;
double d = f;
System.out.println(d);
```



→ 3.4

Quiz 4:


```
double d = 3.4;
float f = d;
System.out.println(f);
```

→ error

```
double d = 3.4;
float f = (float)d;
System.out.println(f);
```

← Correct Syntax

//Operation

↳ Rule 1: Mathematical operation between decimal and non-decimal, Result: decimal

↳ Rule 2: Operation between same Category but different Capacity, Result: bigger size

En: int + long → long

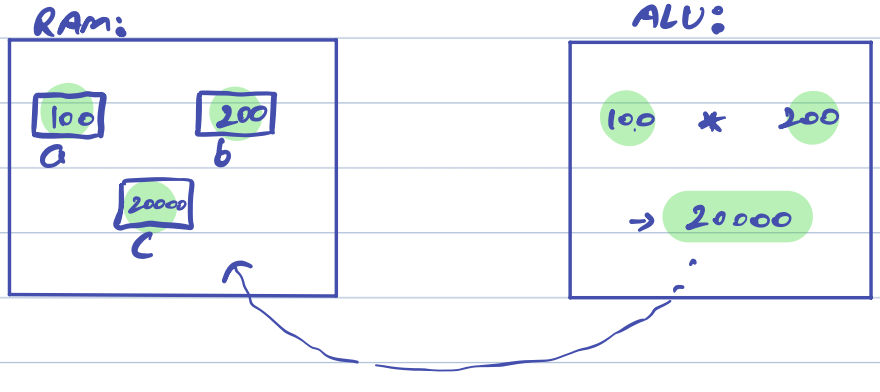
float + double → double

long + double → double

int + int → int

// En:

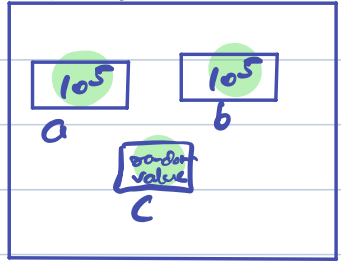
```
int a = 100;
int b = 200;
↳ int c = a*b;
  System.out.println(c); → 20000
```



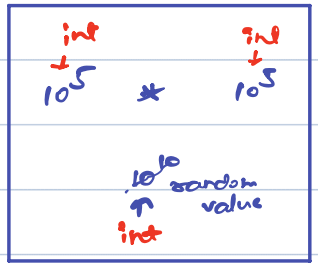
Quiz 5:

```
int a = 100000;
int b = 100000;          → random value
int c = a * b;
↳ System.out.println(c);
```

RAM:



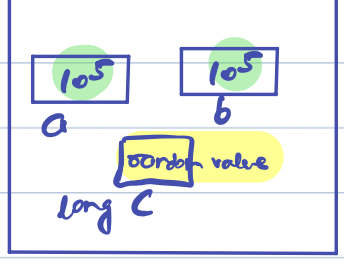
ALU:



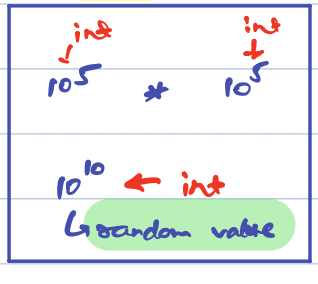
Quiz 6:

```
int a = 100000;
int b = 100000;          → random value
↳ long c = a * b;
System.out.println(c);
```

RAM:



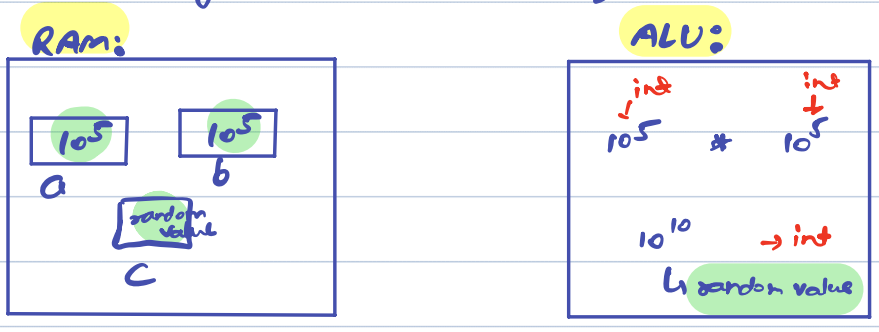
ALU:



Quiz 7:

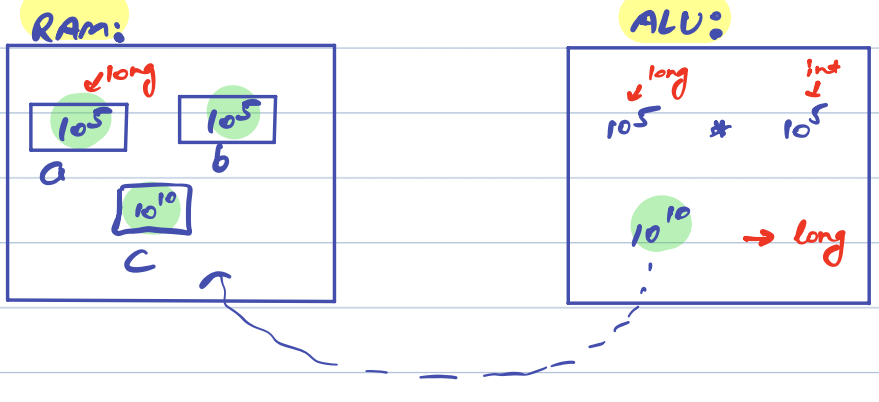
```
int a = 100000;
int b = 100000;          → random value
long c = (long)(a * b);
System.out.println(c);
```

(typecast → long; Calculation → int)



Quiz 8:

```
int a = 100000;
int b = 100000;          → 10^10
long c = (long)(a) * b
↳ System.out.println(c);
```



Break till 9:31 PM

* Arithmetic operators

↳ +, -, *, /, % (remainder)

↳ 20 % 3 : 2

↳ 50 % 10 : 0

Quiz 9:

System.out.println (16/3); → 5

ALU:

16 (int) / 3 (int)

5.333... → int

↳ 5

Quiz 10:

System.out.println (25.0/3);

↳ 8.333...

ALU

25.0 (double) / 3 (int)

8.333... → double

Quiz 11:

System.out.println (35 % 9); → 8

* Relational operators -> used to check relation between 2 data.

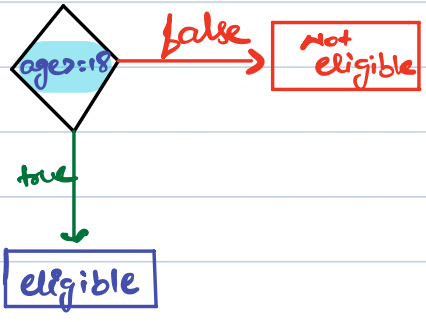
ex: >, >=, ==, <etc.

	x=8 y=10	x=15 y=7	x=13 y=13
x less than y: x<y	true	false	false
x greater than y: x>y	false	true	false
x greater than equal to y: x>=y	false	true	true
x smaller than equal to y: x<=y	true	false	true
x equal y: x==y	false	false	true
x not equal y x!=y	true	true	false

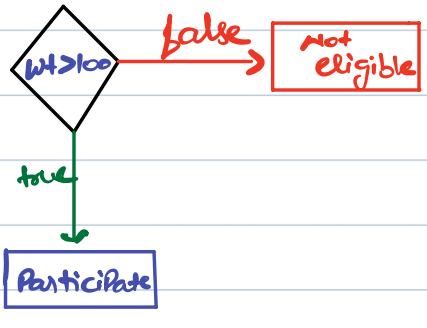
Conditional Statement

// If

1. Check if Person is eligible for car driving license.



2. if Person is above 100 kg in weight, he can participate then only.



idea: when we want to do something on the basis of condition being true.

Syntax:

only cond^n that will return true/false can be written.

```
if (cond^n) {
    // lines you want to execute
    // if the cond^n is true.
}
```

a) 4 == 5 ✓ (valid cond^n but will give you false)

b) 4 < 5 ✓

c) 4 + 5 ✗✗

Q) Read a number (age of person), if person is eligible to get driving license Print "eligible" otherwise don't do anything.

```
Scanner scn = new Scanner(System.in);
int age = scn.nextInt();
if (age >= 18) {
    System.out.println("eligible");
}
```

Quiz 12:

```
int n=20;
if (n>=15){
    S.O.P("Hello");
}
S.O.P("Hello");
```

20
n

Hello Hello

Quiz 13:

```
int n=20;
int y=25;
if (n>=25){
    S.O.P("AlgoPrep1");
}
if (y>=25){
    S.O.P(AlgoPrep2);
}
```

AlgoPrep2

Quiz 14:

```
if (10>6){
    S.O.P("AlgoPrep1");
}
if (15>25){
    S.O.P("2nd");
}
```

AlgoPrep1

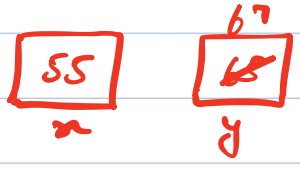
Quiz 15:

```
int x=55;
int y=65;

if (x>55){
    S.O.p ("first");
    x=x+2;
}

if (y>=60){
    S.O.p ("second");
    y=y+2;
}

S.O.p(x+y);
```



false

true

Second 122