Today's agenda

- float and double
- operators
- if else

// Decimal → 2.2, 3.1, 10.4, 12.2 .....

↳ float & double

float → int (6-7 decimal places)

double → long (14-15 decimal places)

```
double temp = 10.2;
```

→ Integer, default value is int.

→ decimal, default value is double.

Integer → `long temp = 10^10L;`

decimal → `float f = 2.4f;`

* 2 Golden Rules of typecasting

↳ 1. If there is guaranteed no loss of data : implicit

ex: int to long

float to double

↳ 2. If there is a chance for loss of data then,

Conversion is Possible but forcefully. explicit

Ex: long to int

double to float

Quiz 1:

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

2.8 d

Quiz 2:

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

3.3 f

Quiz 3:

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

3.4 f 3.4

Quiz 4:

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

3.4 d

```
double d = 3.4;
float f = (float)d;   <- correct syntax
System.out.println(f);
```

//Operation

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

↳ Rule 2: Operation between same category but different capacity, result: bigger size

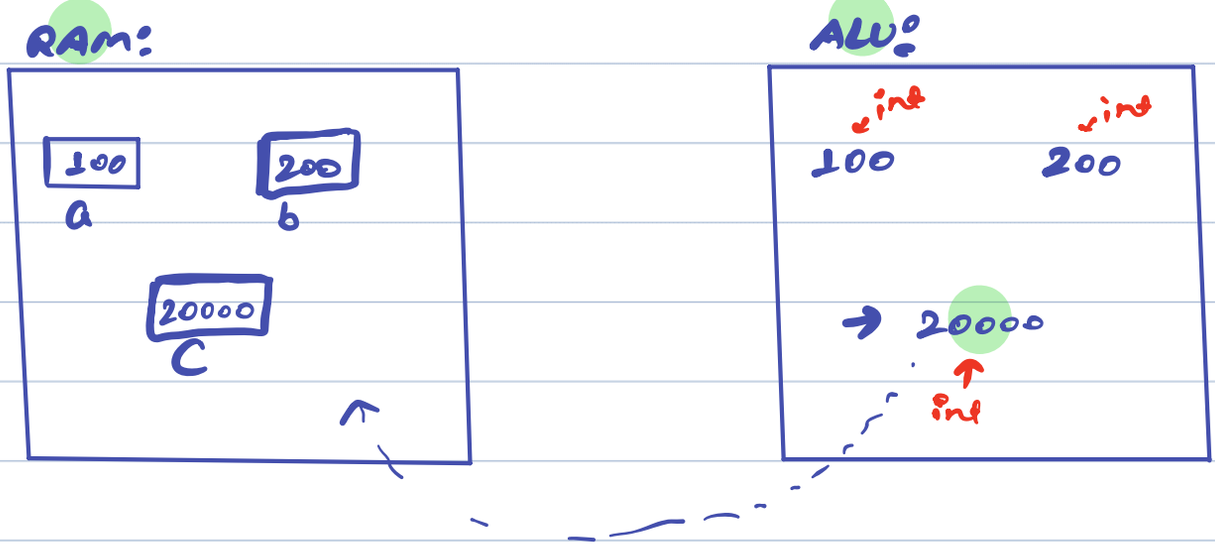
int + long → long

float + double → double

long + double → double

// Ex:

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

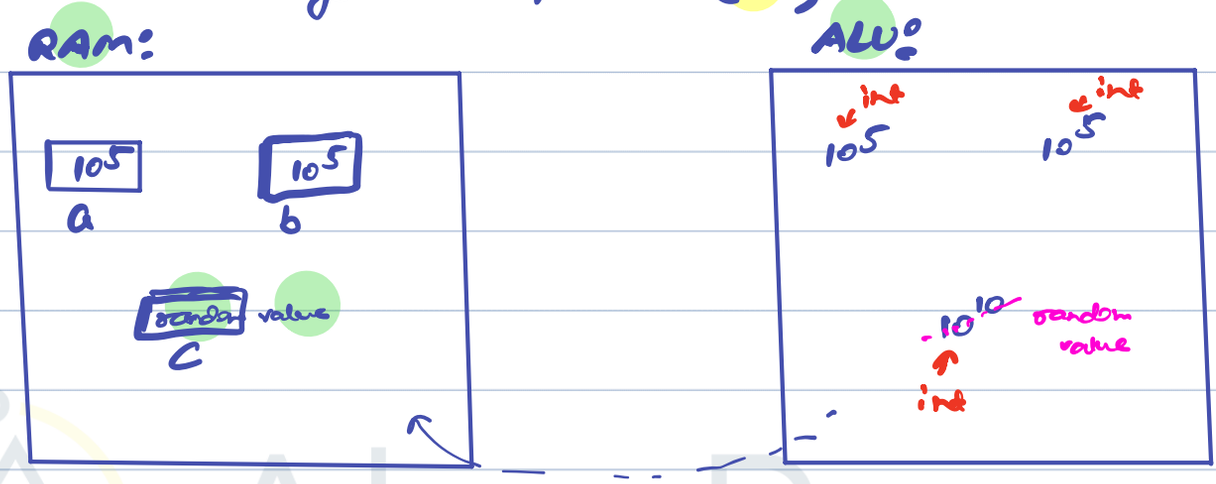


int → {$-2*10^9$, $2*10^9$}

long → {$-2*10^{18}$, $2*10^{18}$}

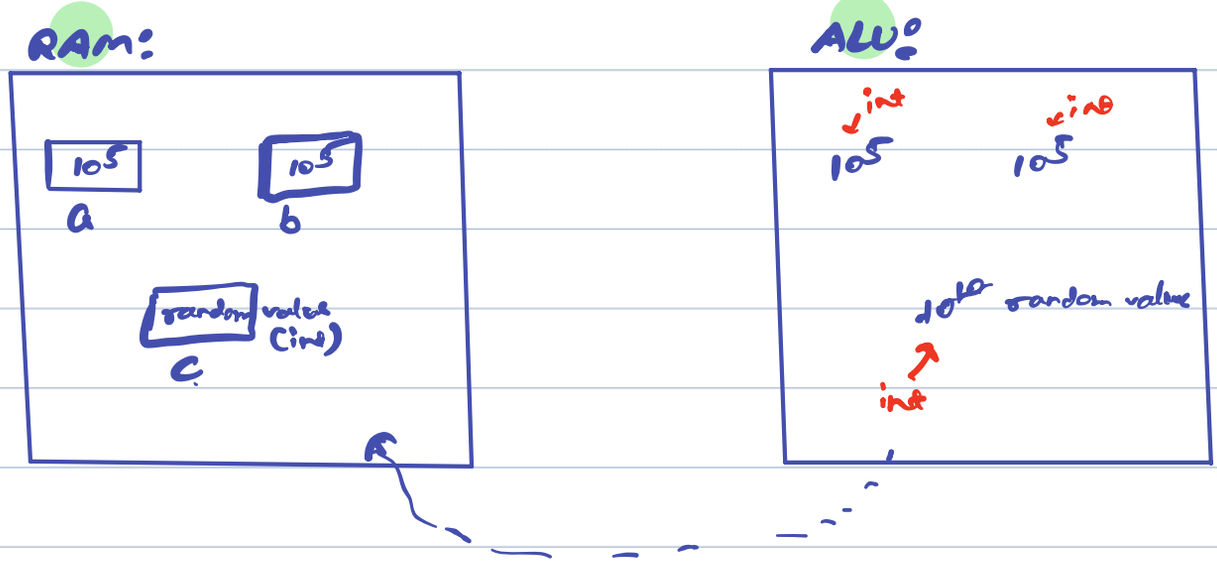
Quiz 5:

```
int a = 100000;
int b = 100000;
int c = a*b;
System.out.println(c);
```



Quiz 6:

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

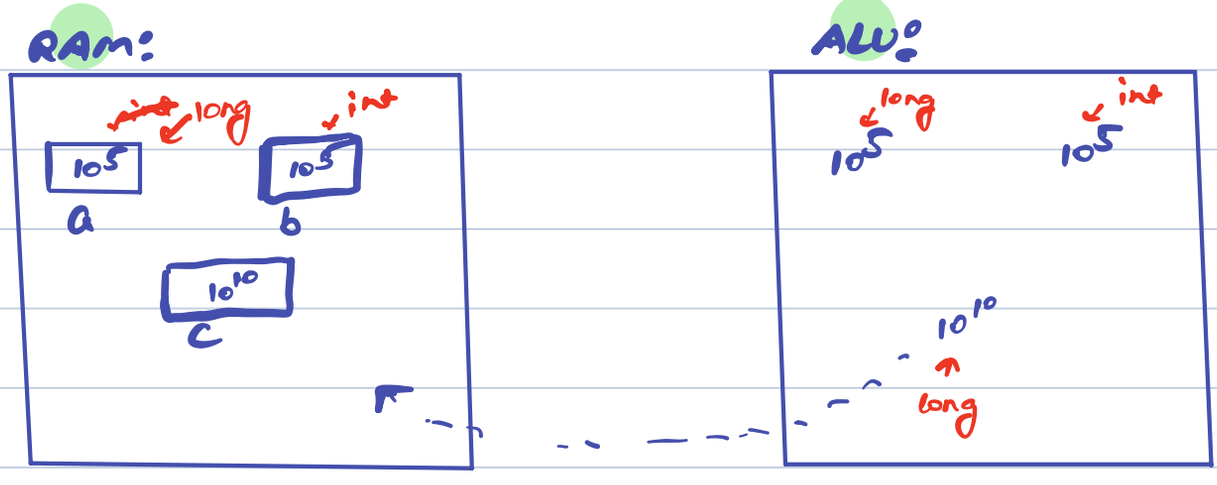


Quiz 7:

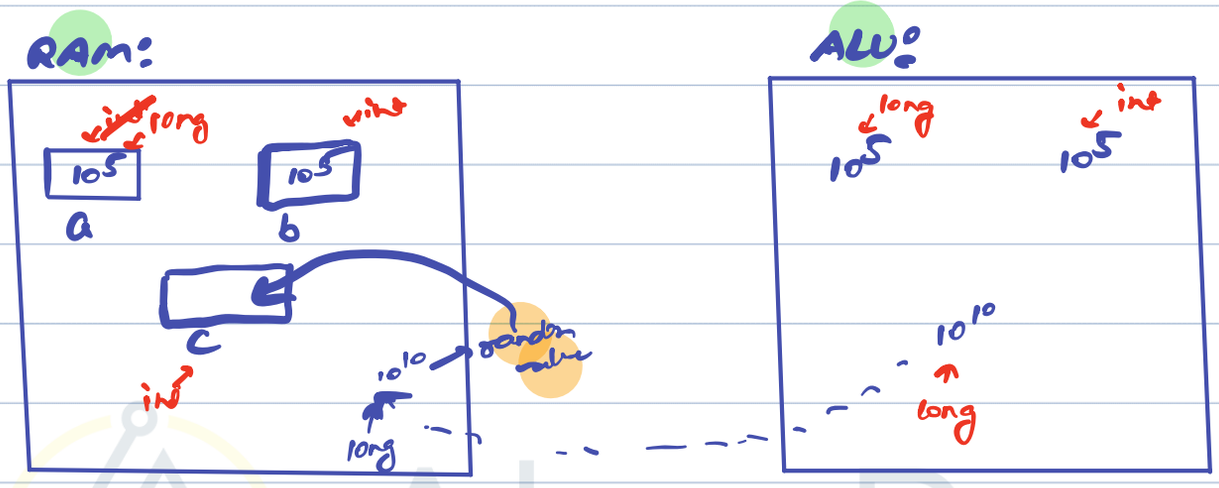
```
int a = 100000;
int b = 100000;
long c = (long)(a * b);
System.out.println(c);
```

Quiz 8:

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



```
int a = 100000;
int b = 100000;
↪ int c = (int)[(long)(a) * b];
System.out.println(c);
```



Break till 9:53 PM

* Arithmetic operators

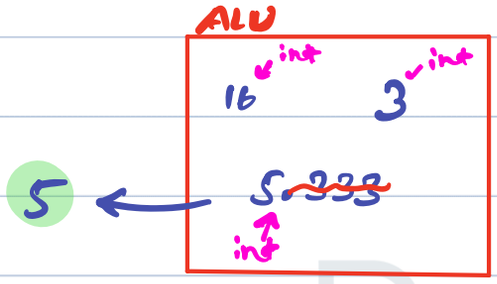
↳ +, -, *, /, % (remainder)

↳ 20 % 6 = 2

↳ 25 % 5 = 0

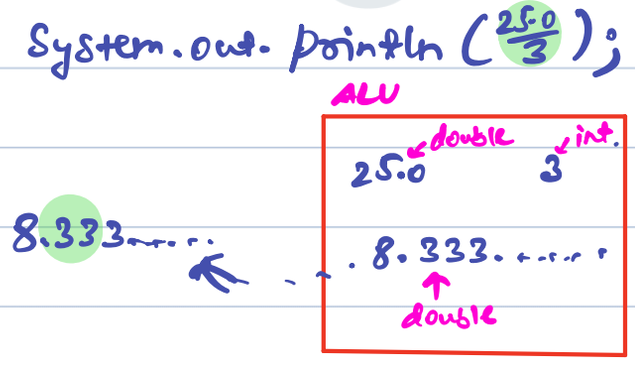
Quiz 9:

System.out.println(16/3);



Quiz 10:

System.out.println(25.0/3);



Quiz 11:

System.out.println(35 % 9); → 8

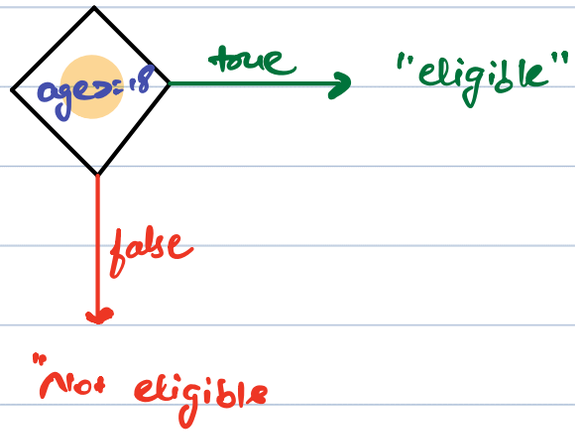
* Relational operators → used to check relation between 2 data.
ex: >, >=, == etc.

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

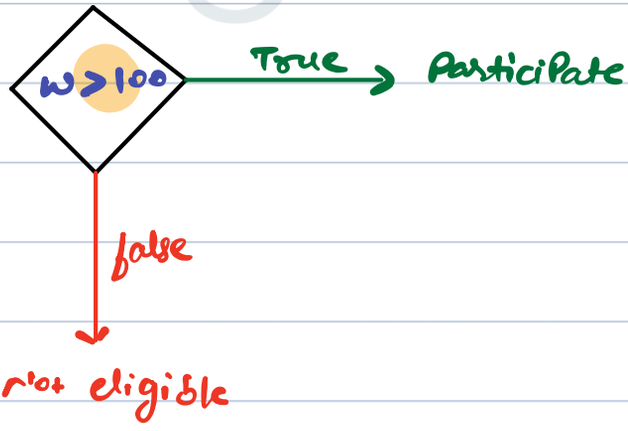
Conditional Statement

// 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:

Cond^n should be relational. ↳ true/false

↳ 4 == 5 ✓

↳ 5 < 3 ✓

↳ 5 + 3 ✗✗

```
if (Cond^n) {
    // lines you want executed
    // if Cond^n is true.

}
```

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


```java
// "static void main" must be defined in a public class.
public class Main {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int age = scn.nextInt(); //15

        if(age >= 18){
            System.out.println("eligible");
        }


    }
}
```

stdin

15

Share Live Add Snippet

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){          // false
    S.O.p("AlgoPrep1");
}
if (y>=25){          // true
    S.O.p("AlgoPrep2");
}
```

20 (n) 25 (y)

→ AlgoPrep2

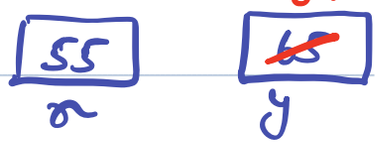
Quiz 14:

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

AlgoPrep1

Quiz 15:

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




```
if (x>55) {        // false
    S.O.P ("first");
    x = x+2;
}
```

second 122

```
if (y >= 60) {     // true
    S.O.P ("second");
    y = y+2;       // 67
}
```

→ `S.O.P(x+y);`