Today's agenda

- float and double
- Taking input
- operators
- if else

// Decimal → 2.2, 1.3, 4.7, 2.6 etc.

↳ float & double → long (. 14-15 decimal places)

float → int (. 6-7 decimal places)

→ In Integer, default values are int

→ In decimal, default values are double.

Writing float

↳ float a = 2.4f;

Writing double

↳ double d = 4.2;

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:

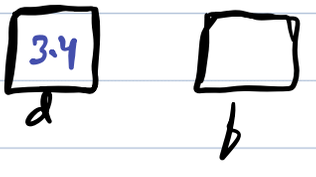
```
float f = 3.4f;
↳ double d = f;
System.out.println(d);
```

→ 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);
```

2 Golden rules of typecasting

1. If there is no loss of data then no error: implicit
2. If there is chance for loss of data then error.
   We can still do this change forcefully: explicit (typecasting)

- int to long → works
- float to double → works

- long to int → explicit
- double to float → explicit

Operation

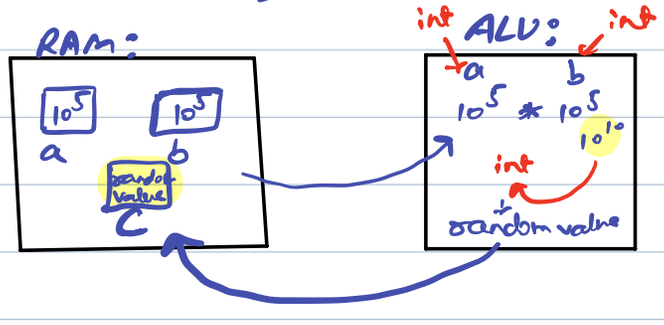
↳ **Rule1:** Mathematical operation between decimal and non-decimal, Resultant: Decimal

↳ **Rule2:** Operation between same category but different capacity, Resultant: higher size

a	b	res
int	int	int
int	long	long
long	int	long
long	long	long

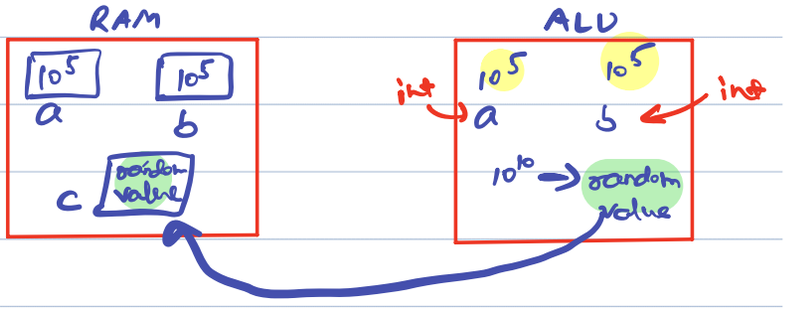
Quiz5:

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



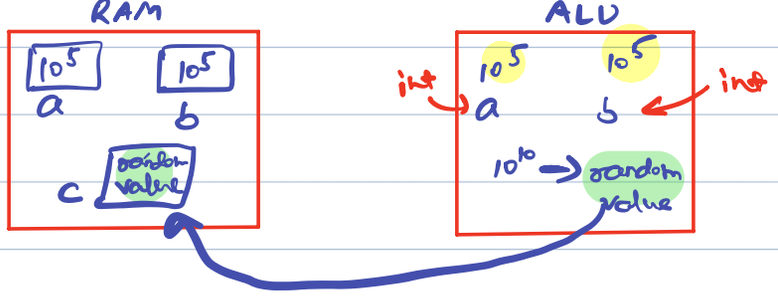
Quiz 6:

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



Quiz 7:

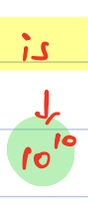
```
int a = 100000;
int b = 100000;
long c = (long)(a*b);
System.out.println(c);
```

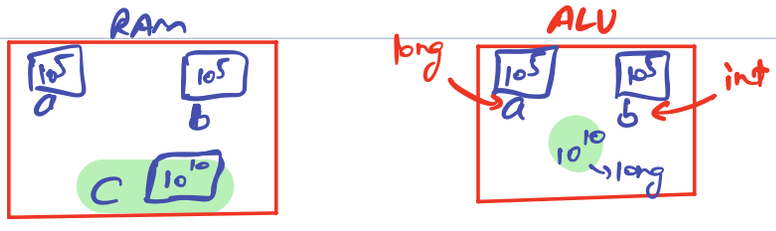


Quiz 8:

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

→ b is correct





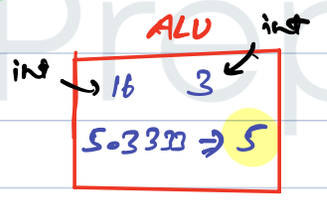
* Arithmetic operators

↳ +, -, *, /, % → remainder

↳ 20%3 = 2

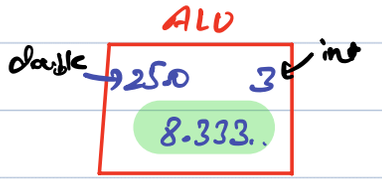
Quiz 9:

System.out.println(16/3); → 5



Quiz 10:

System.out.println(25.0/3); → 8.333...



Quiz 11:

System.out.println(35%9); → 8

↳ x % 7 → {0 ..... 6}

x % N → {0 ..... N-1}

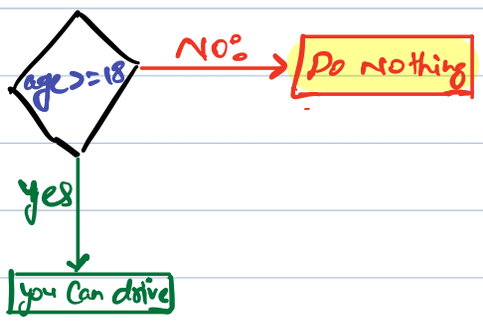
Break till 9:25

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

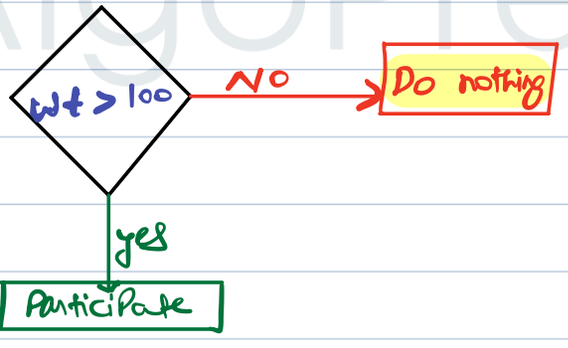
	x=8 y=10	x=15 y=7	x=13 y=13
x less than y : x<y	true	false	false
x greater than y : x>y	false	true	false
x greater than equal to y : x>=y	false	true	true
x smaller than equal to y : x<=y	true	false	true
x equal y : x==y	false	false	true
x not equal y : x!=y	true	true	false

// If

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



2. Check if Person is above 100 kg in weight. he Can Participate in weight lifting.



idea: when we want to do Something on the basis of Condition being true.

Syntax:

```
if (condn) {
    // statement or lines you want to
    execute if the condn is true.
}
```

→ only condn which will give you true/false. (Relational OR)

a) 4 == 5 ✓ (valid but give you false)
b) 4 < 5 ✓

c) 4+5 xx

Q) Read a number → age of person and if Person is eligible, Print "eligible for driving licenle."

//Pseudo code

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

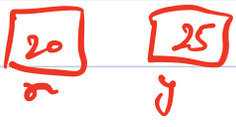
Quiz 12:

```
int n= 20;
if (n >=15){
    System.out.print( " Hello");
}
System.out.println( " Hello");
```

Hello Hello

Quiz 13:

```
int n= 20;
int y= 25;
```




```
if ( n >= 25) {
    System.out.println ("Algoprep1");
}
if (y >= 25) {
    system.out.println( "Algoprep2");
}
```

→ Algoprep2

Quiz 14:

```
if (10 > 6) {          → true
    System.out.println("1st");
}
if (15 > 25) {         → false
    System.out.println("2nd");
}
```

1st

Quiz 15:

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

if (x > 55) {          → false
    System.out.print("first");
    x = x + 2;
}
if (y >= 60) {         → true
    System.out.print("second");
    y = y + 2;
}
System.out.println(x+y);
```

x: 55

y: 67 (~~65~~)

second 122

65 + 2 = 67