Class 1: Hello World !!!

Print

→ System.out.println ("Hello World");

→ (;) → one line of code has ended.

→ System.out.println ("Elephant");

next line
↳ brings the cursor to next line.

Data types

- → [Primitives] → hold only a single type of value
- → Non-primitives.]ˣ

Variables

int (a) = 5

datatype

[5] a,b,c, cat, dog.

'ac'

['a'] 'b'

char

→ 8 types of primitive datatypes

-128 to 127   [(byte)   int]
              [short   long]

                  Integer

float
double
decimal value.

boolean
↓
true/false

0) Float , Boolean
            print

How to take input in Java.

Scanner sc = new scanner (System.in);

'a'

[or Kunal is] 8 5
→ String

Non primitive
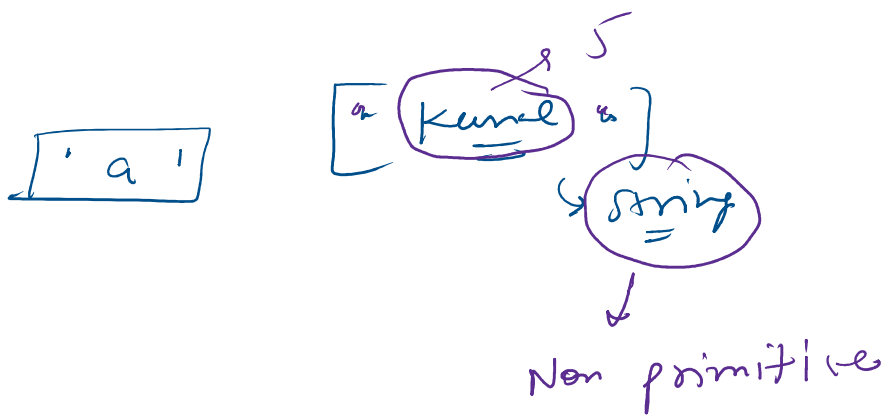
String + ⊘⊘⊘⊘ ⟹ String

Hello + 5
string + int ⟹ string] Type casting

Basic Mathematical operators

→ + ⟹ 5+5 = 10
→ − ⟹ 7−5 = −2
→ * ⟹ 5*2 = 10

↗ → quotient

/ → divide] → quo
% → modulo
      remainder

→ * ⇒ 5 * 2 = 10

→ /

→ %

⇒ quotion

5) 13
   2

   10
   ---
   1 → remai

remainder

13 / 5 = 2

13 % 5 = 3

$$\text{int } a = 5;$$

declaretion ← → initialization

e

e

→ String + int ⇒ string] Type casting

Type casting
   is process of converting one data type into another

→ [implicit] → [apne aap]
→ [explicit] ←

int + flot

double

byte → short → int → long → float → double
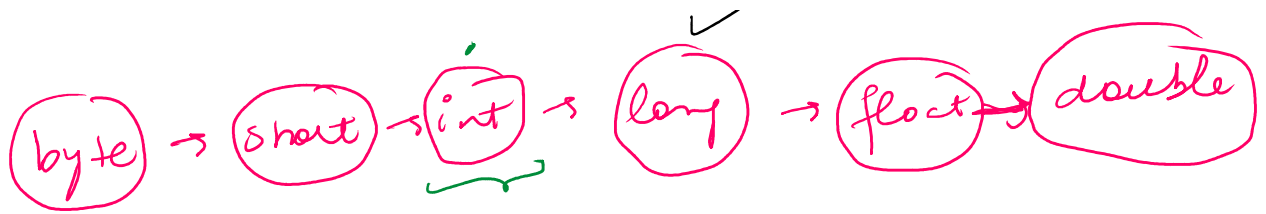
long d = 5000

int e = 10000

String + double

5.3 + 5
double + int

Relational operators → boolean

==

>= <= > < 10 > 5

→ 10 > 5   → 10 > 5   → 11 <= 10

↳ 5 < 3   → 5 >= 5   → 10 <= 10

age >= 18 → allowed
→ not allowed

Conditionals

[if - else]

if (condition) {

}

if (age >= 18) {
  sout (" allowed")
}
else {
  sout (" Not allowed)
}

else ;
‗
‗

else ;
sout (~ Not allowe)
;

$=$ → assignment
$==$ → comparison

comparison
↓
boolean

[$b = 50$
$b == 5$] true

b
[5]

↳ if can exist with out else

[if (~) ;
‗
;] ✓

~~else~~ ✗
‗

if- else ladder

if-else ladder

```
if ( ~ ) ?
    )
else if ( ~ ) ;

    else if ( ~ ) { } ]
    :
    else {      }
```

6) Num input
   'Evens, Odd' ?

Ⓑ Not equal   !=

         5 != 3 → True   10 != 10