"समय न लगाएँ इसमें कि क्या करना है,
वरना समय ये तय करेगा कि आपका क्या कराना है।"

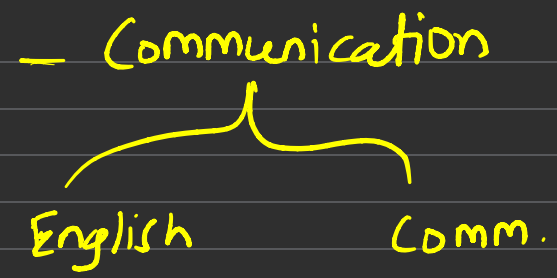
Welcome & Vision

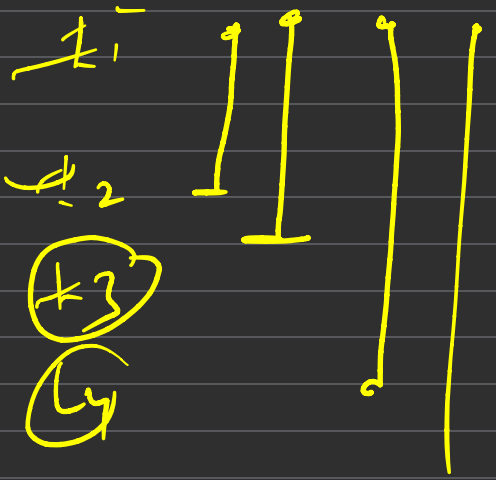
Why you started this course.

"I don't just want you to learn coding.....

Don't Mug up - You are Engineer

Change Studying methods





Course Roadmap

Roadmap – Java – OOPs – Collection Frame – DSA

90%

HomeWork Problems

Class Code

Notes

DSA
OOPs
DB
Project
Aph

2 of

first sem = DSA / Aph / Mp

project = React JS

Let's Start the JAVA 🤩

What is Programming ?

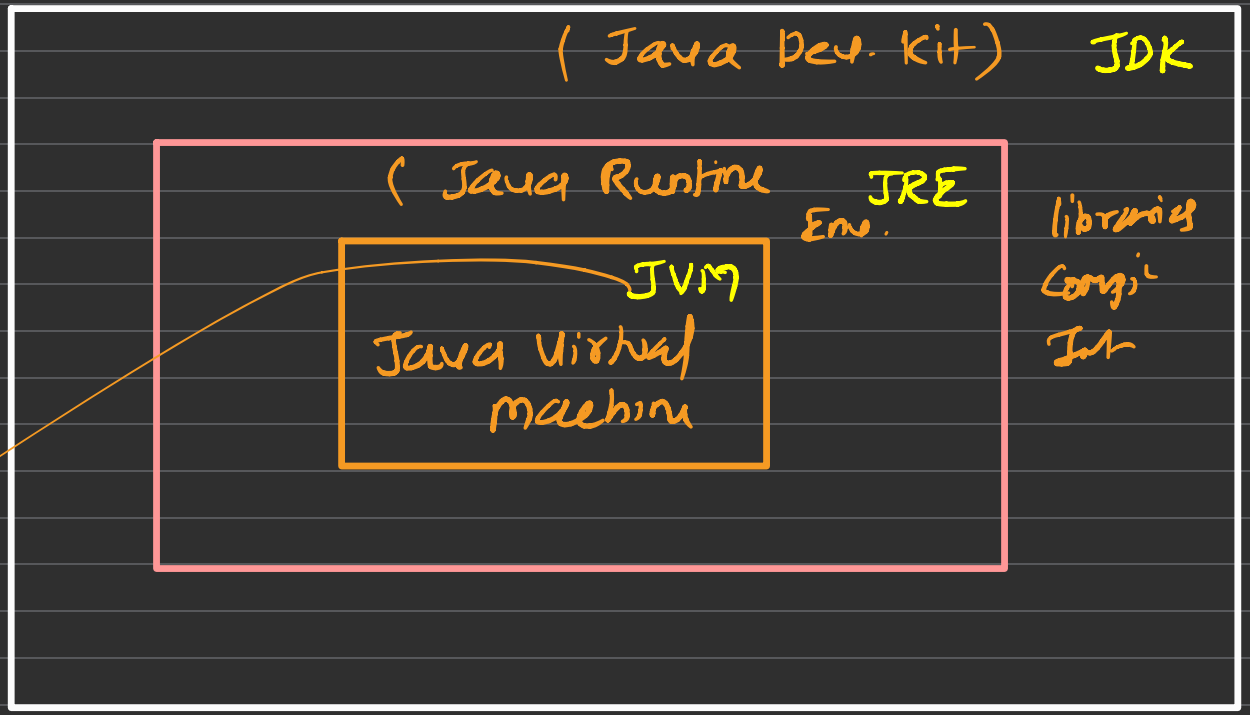
Programming means giving instructions to a computer to do something for us.

What is JAVA Programming ?

- OOP → mainly used to build Enterprise Application (Banking 90%)
- 1995 — James Gosling

JDK JRE JVM

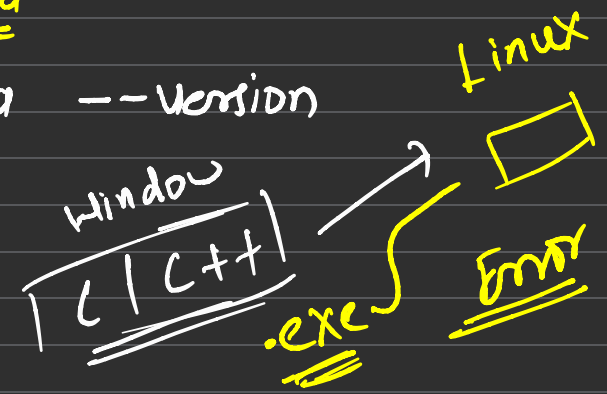
Kitchen + Chef + Recipe



Setting Up Java Environment

Window = Wind + R = cmd

java --version



VVIMP

Features of Java Programming

* OOP * Platform Independent * Secure

4 Portable — Window — Linux — Mac

AB int

4B (32 bit)
2B (16 bit)

Structure of JAVA Programming

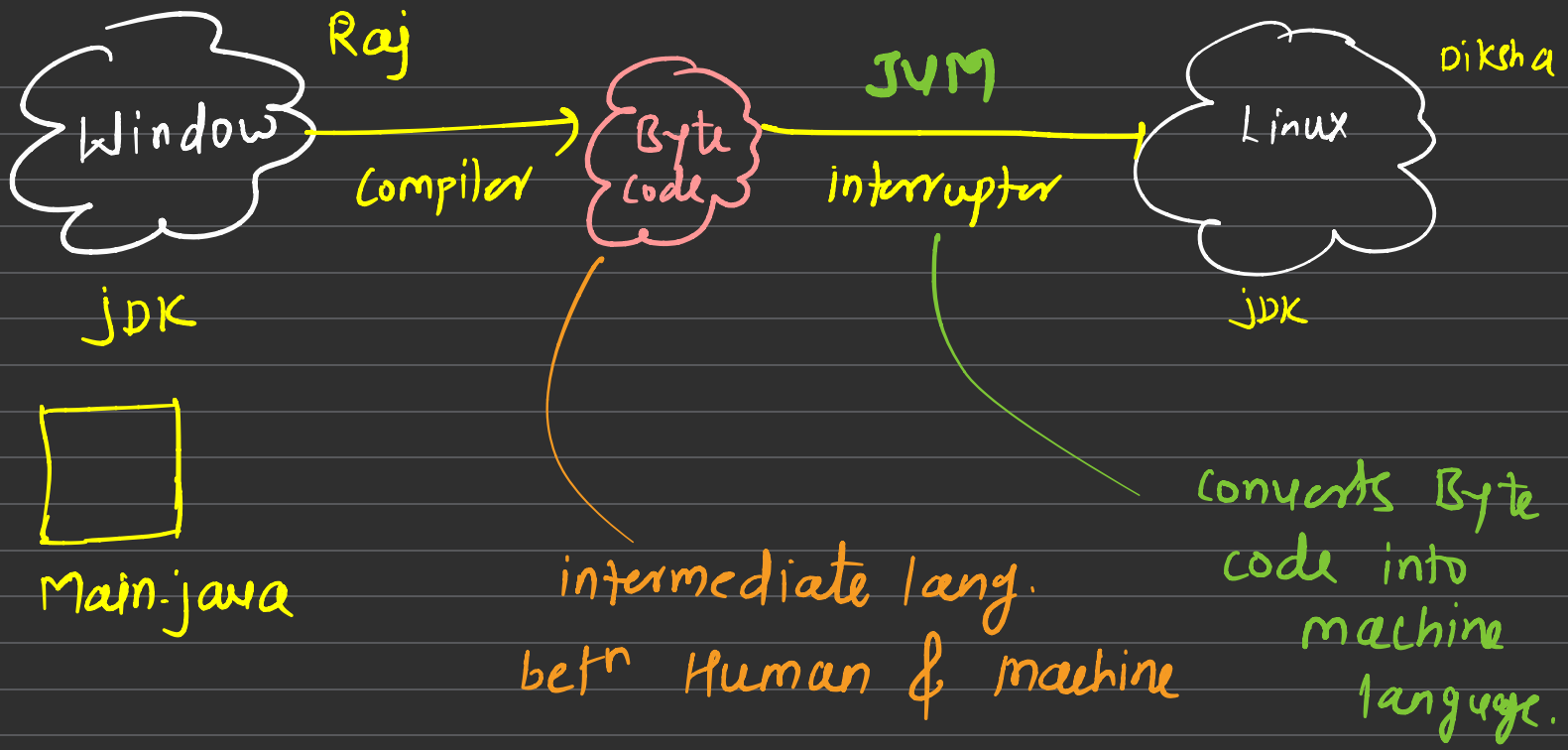
OOP (class & object)

```java
public class Main {
    public static void main(String[] args) {

    }
}
```

Class Name and File Name should be same

class which contain main function → must be public & class Name = fileName.java



public class Main
{

psvm ( )
{

}

}

class Hello
{

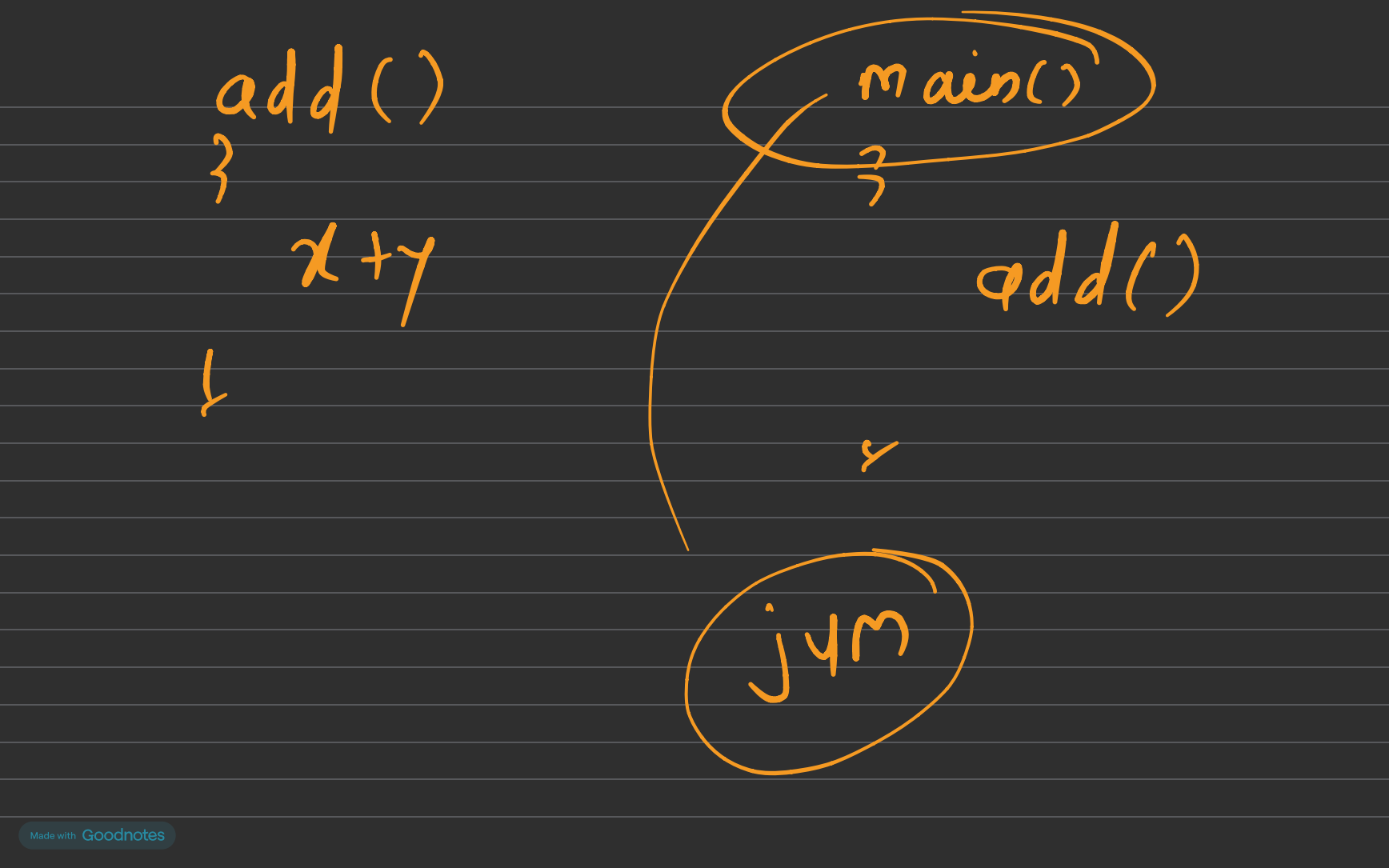
}

javac Main.java (compile)

main.class

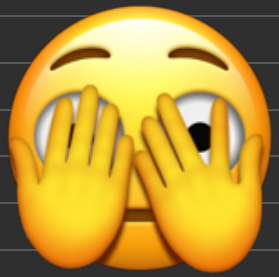
Hello.class

java Main (run)



Writing First “Hello World” Program

How to Print ?

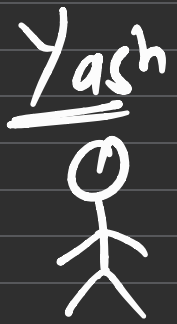


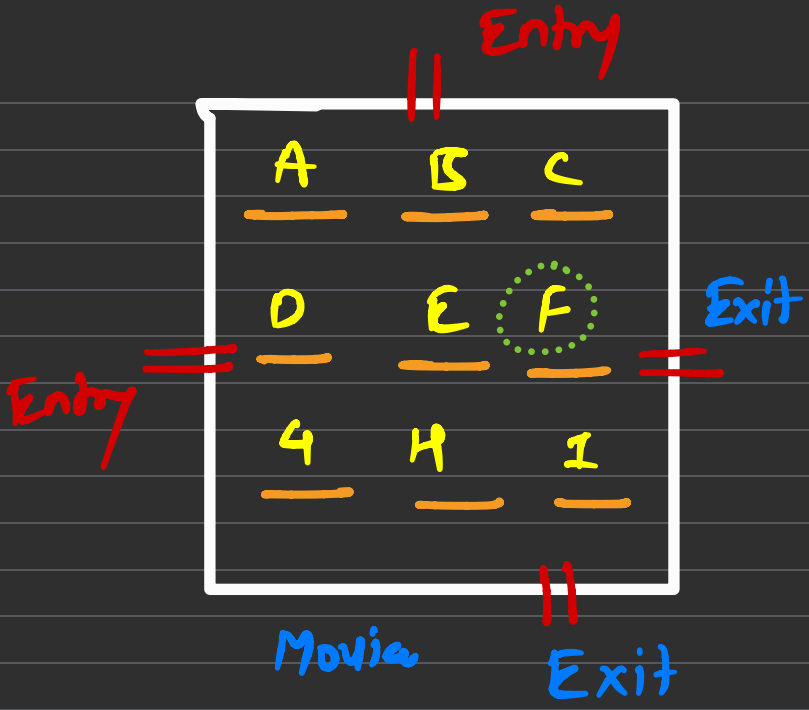
What is System.out.println

System.out.println

Variables & Data Types

What is Variable

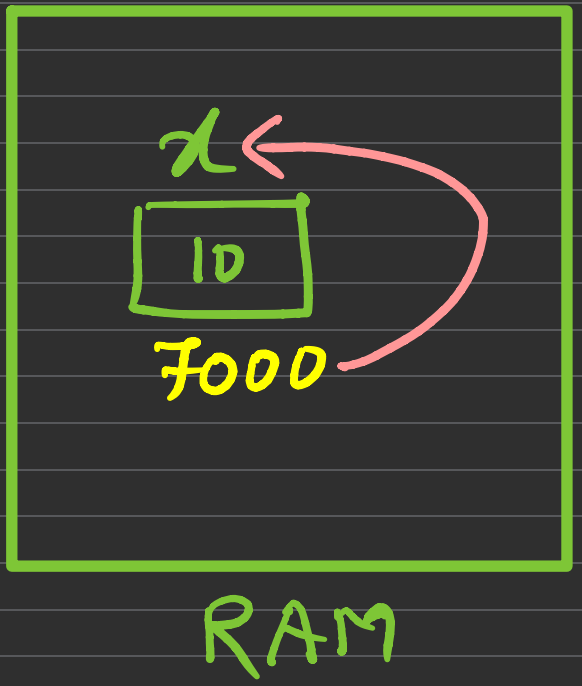




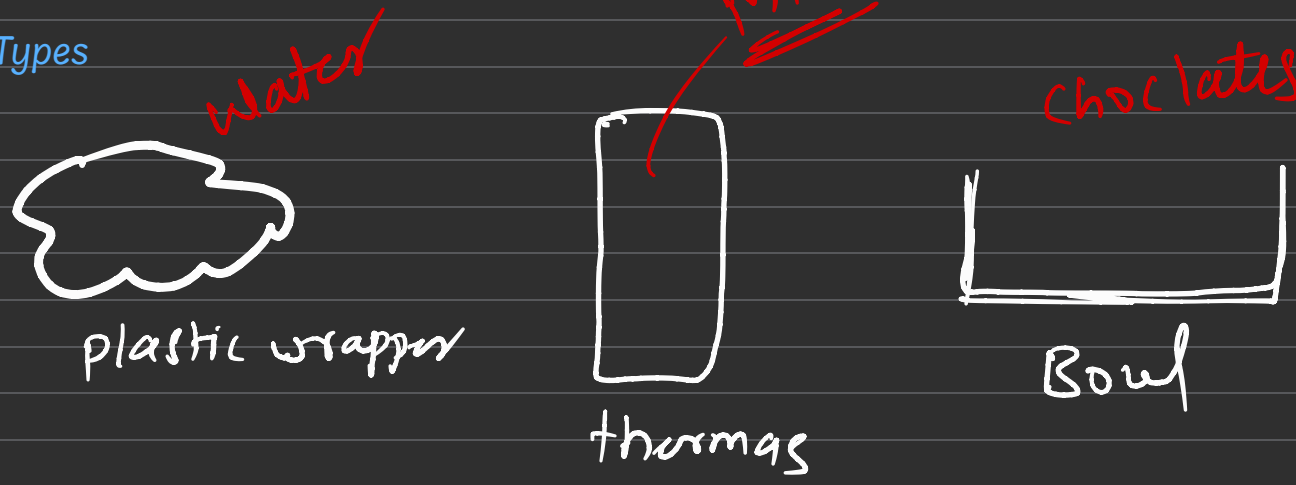
xyz

```
int x = 10
print(x)
```

Name given to memory location

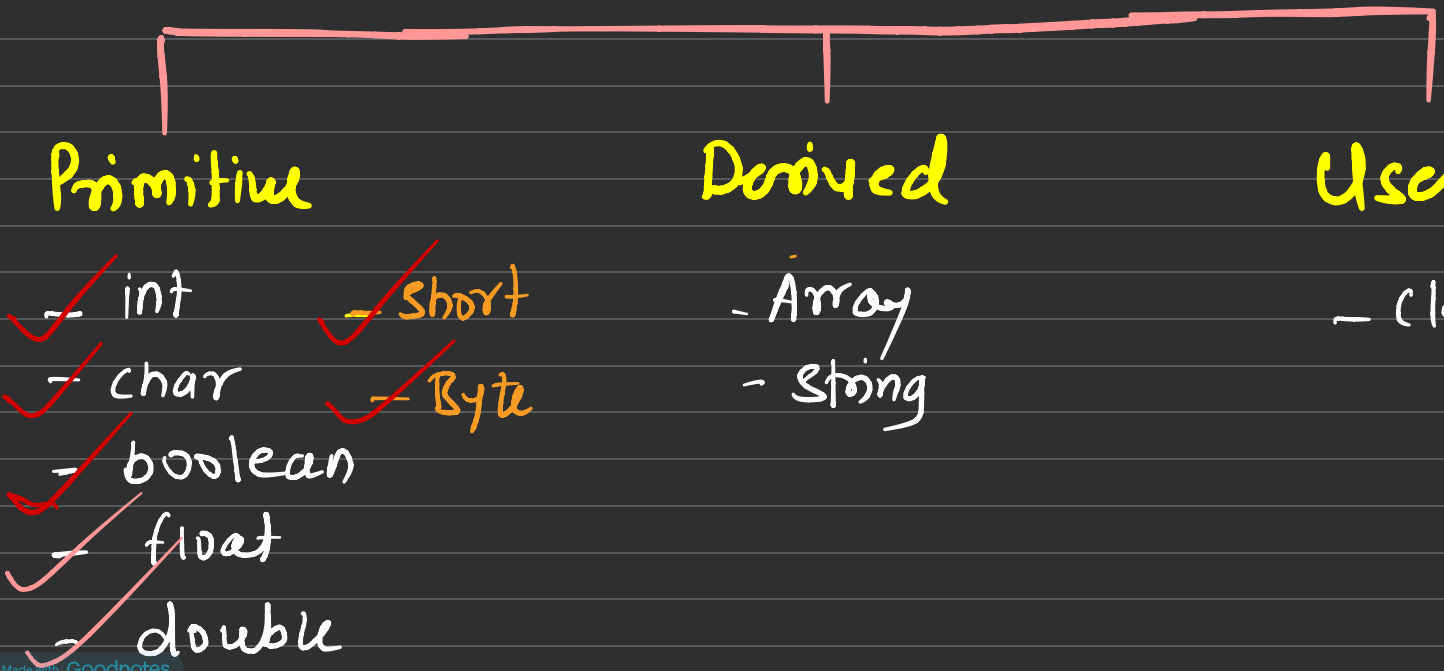


What is Data Types



- Mainly tells what type of data we store.
- Also tells us, what type of operations we can perform on them

Types of Data Types



(int) = mainly stores decimal values

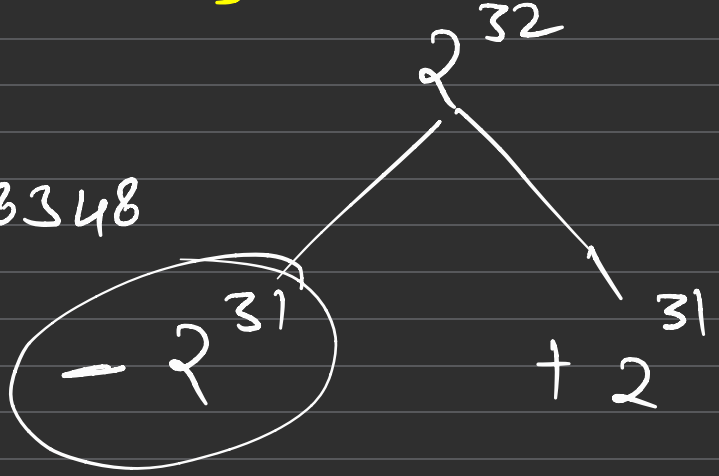
size = 4B 1B = 8 bits

4B = 32 bits

int x { Pos, Neg }

$\frac{2^{32}}{2^1} = 2^{32} \times 2^{-1}$

2^{31}



-2147483648

2147483647

Range = -2^{n-1} — $2^{n-1} - 1$

Data Type

int / char / short
Byte / long

(n = No. of bits)

- short (int)

Size = 2B 1B = 8 bits = 16 bits

Range = $-2^{15} - 2^{15}-1$ = -32768 to 32767

- Byte (int)

Size = 1B 1B = bits

Range = $-2^{7} - 2^{7}-1$ = -128 to 127

long (int)

size = 8B 1B = 8 bits

- boolean
  - True
  - False

boolean x=

- Char

  - Char are shown / declared in Single quote

Eg: 'x' '+' '/' 'z' '?'

size = 2B

(

unicode

C / C++ = ASCII values

(-128 to 127)

(0 to 255)

Size = 2B

1B = 8 bits

Range = 0 ⟶ 65535

java

(0 - 255) = ASCII

(256 — 65535) = Unicode

float / double = floating points

(2.16 / 3.09)

Java

float y = 90.8F

double x = 90.6

float x = 90.9D

(CPP)

double y = 90.8

How to take input from the user

```
import java.util.Scanner
```

Scanner = mainly scans the input given by User through Keyboard

```
Scanner sc = new Scanner(System.in)
```

sc → object

```
Scanner sc = new Scanner(System.in)
```

* int

```
int x = sc.nextInt()
```

* boolean

```
boolean x = sc.nextBoolean()
```

* char

```
char x = sc.next()
```

* double x = sc.nextDouble()

* nextLin()

(

Reads multiple words

with Space

(String)

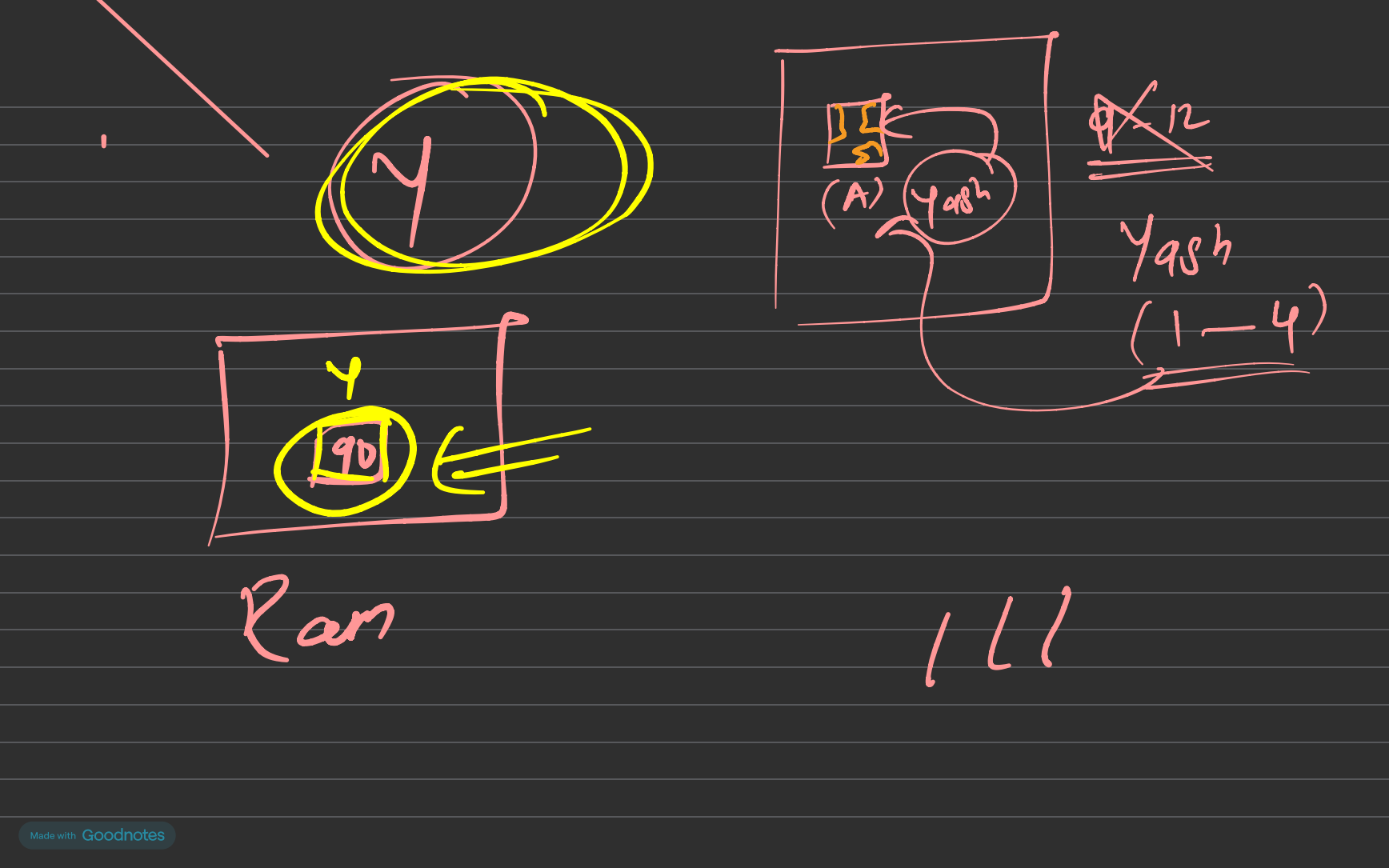
* next()

(Single word)

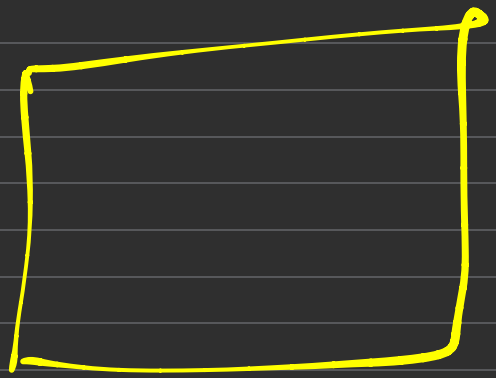
(

String

Method	Work
nextInt()	int
nextBoolean()	Boolean
nextDouble()	Double
nextFloat()	Float
nextLine()	String with multi word
next()	single word



(30 min) Garbage Collector



MM (Ram)

Operator Types

- Unary operator: which req. single operand

  x++ , ++x , --x , x--

- Binary operator: which req. two operand

  x+y , x<y x||y x&&y

- Ternary operator: which req. three operands

  x ? y : z

Operators

Arithmetic Operator

+ − * / %

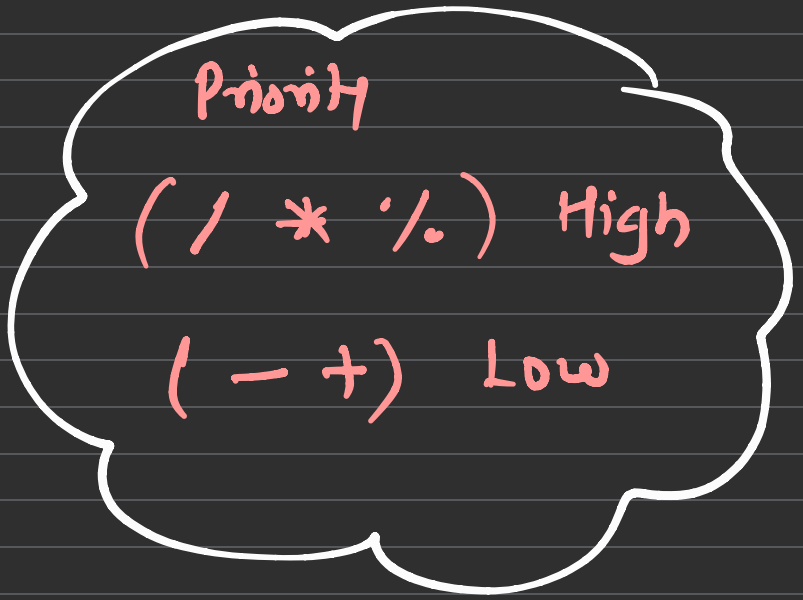
(+) Add

(−) Sub

(*) Mul

(/) div

(%) Reminder



$3 / 4 + 6 - 3 * 2 + 9 \% 2$

$x = 4 \qquad y = 7$

$ans = x * y$

$ans = \underline{28}$ (integer)

Relational Operator

Mainly returns boolean { True, false }

$< \quad > \quad \leq \quad \geq \quad == \quad !=$

Priority

(High) = $< \quad > \quad \leq \quad \geq$

(low) = $== \quad !=$

$$A = 30 > 20 > 0 \;!= 2 < 50 > 40 \;!= 50$$

$$1 > 0 \;!= 2 < 50 > 40 \;!= 50$$

$$1 > 0 \;!= 1 > 40 \;!= 50$$

$$1 > 0 \;!= 0 \;!= 50$$

$$1 \;!= 0 \;!= 50$$

$$1 \;!= 1 \implies 0$$

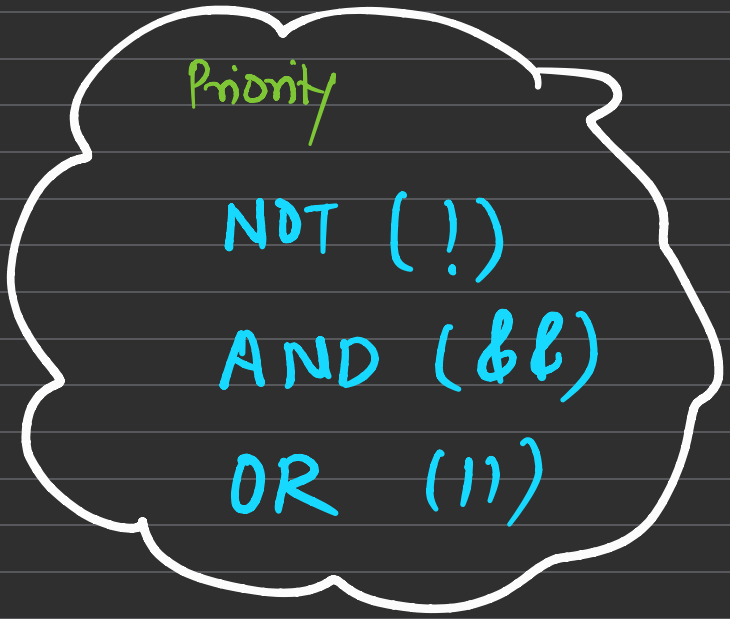
Logical Operator

mainly returns either True or false

OR (||)

AND (&&)

NOT (!)



NOTE

* Any Non-Zero value is True { Pos, Neg }

NOT

!7 → ! True → False

!0 → ! False → True

AND (&&)

(A && B = Y)

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR (||)

(A || B = Y)

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

AND

① x && y

- if x is true then and then only y is checked

- if x is false, then y is not checked

OR (||)

- if x is True, then y is not checked
- if x is false, then y is checked

Increment Decrement

* Increament

- pre-Increment = first increase & then use
- post-Increment = First Use & then increase

- pre-decrement = ==//==
- post-decrement = ==//==

* pre

```
int x = 10
y = ++x
z = x+y
printf(x, y, z)
```

```
int x=10
x=x+1   x=11
y=x     y=11
z=x+y   z=22
printf(x, y, z)
```

* post

```
int x=20
y=x++
z=x+y
pf(x,y,z)
```

```
int x=20
y=x     y=20
x=x+1   x=21
z=x+y   z=41
pf(x, y, z)
```

```
int m = 10
n = ++m
n1 = m++
n--
--n1
n = n - m
pf(n)
```

```
int m=10
m=m+1
n=m
n1=m
m=m+1
n
n=n-1
n1=n1-1
n1
n=n-m
```

m	n
~~11~~ 12	~~11~~ 10

n1
~~11~~ 10

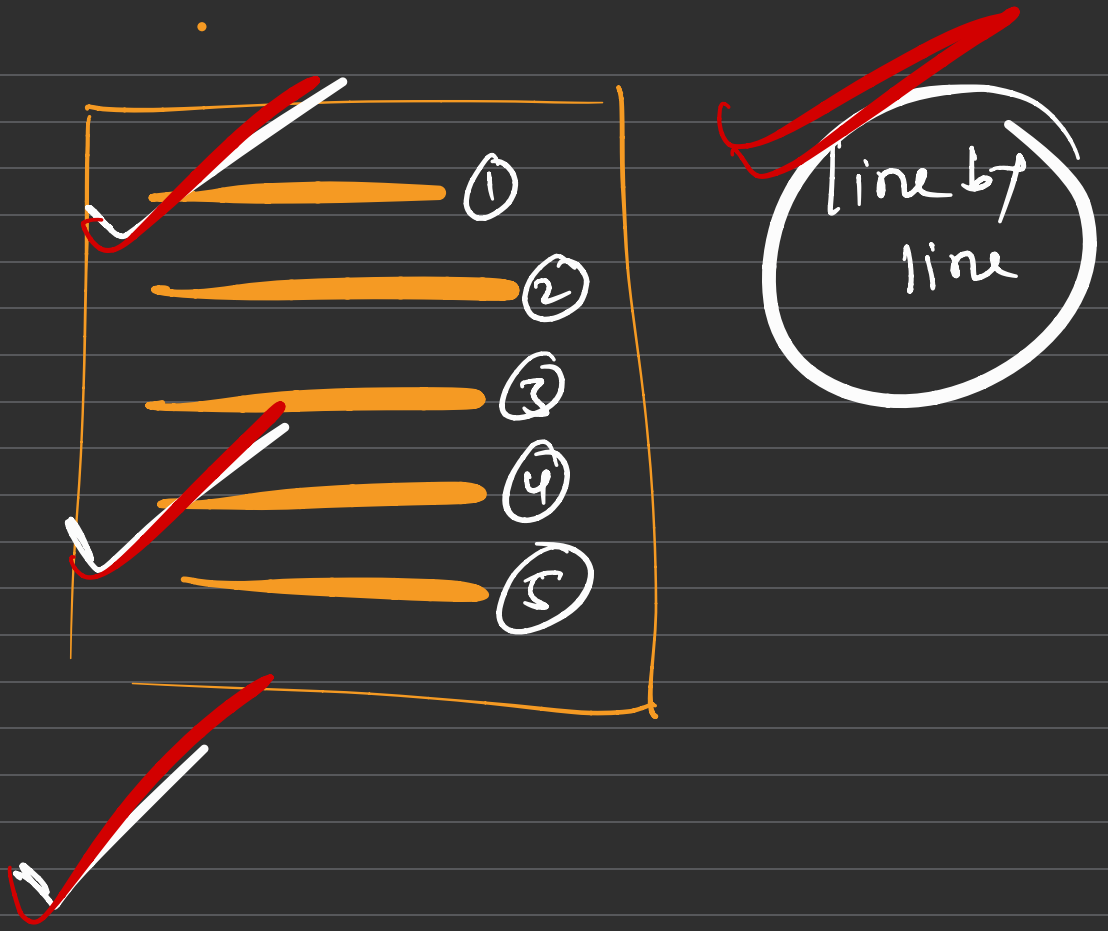
10 - 12

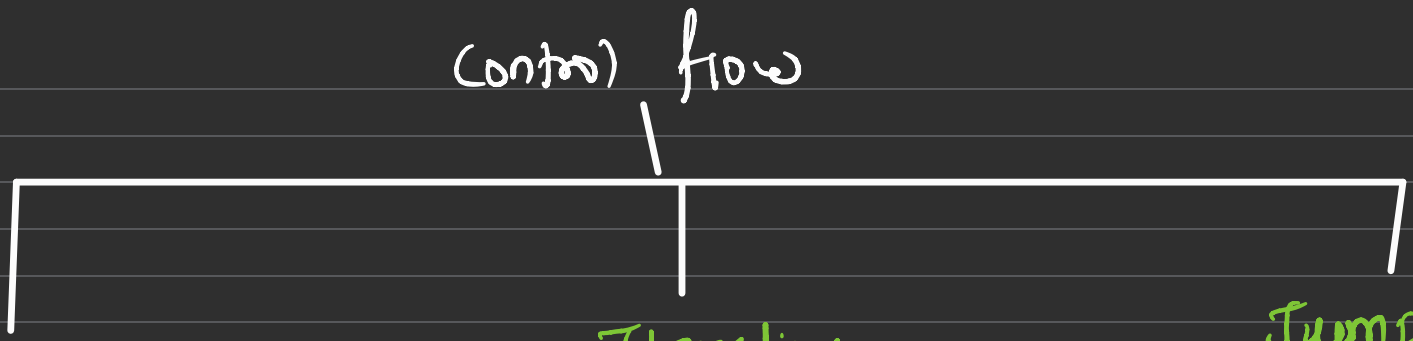
-2

12 | -3 |

Ternary Operator

Control Flow Operator





if :

Syntax:

```
if ( condition )
{
    // statement
}
```

```
if ( 5 )
{
    println( Hello )
}
```

```
if ( Expression )
{
    // statements
}
```

Any statement which has value

```
if (15 < 10)
    println("Hello");
println("Bye");
println("Hi");
```

Bye
Hi

if no curly braces, then only first semi-colon statement is consid

(loop / if / else)

if else

- if is True, then else will not execute
- if is false, else will execute

Syntax:

```
if ( Expression)
{

}

else
{

}
```

```
if (a < 10)
    println("Hello");
else
    println("Bye");
```

```
if (a > 10)
    println("Hi");
else
    println("Bye");
```

Compilation Error

```
x = 10, y = 20
if (x < y)
    println("Hi");
    println("Bye");
else
    println("GN");
```

Loop: mainly used to do the work Repeatedly

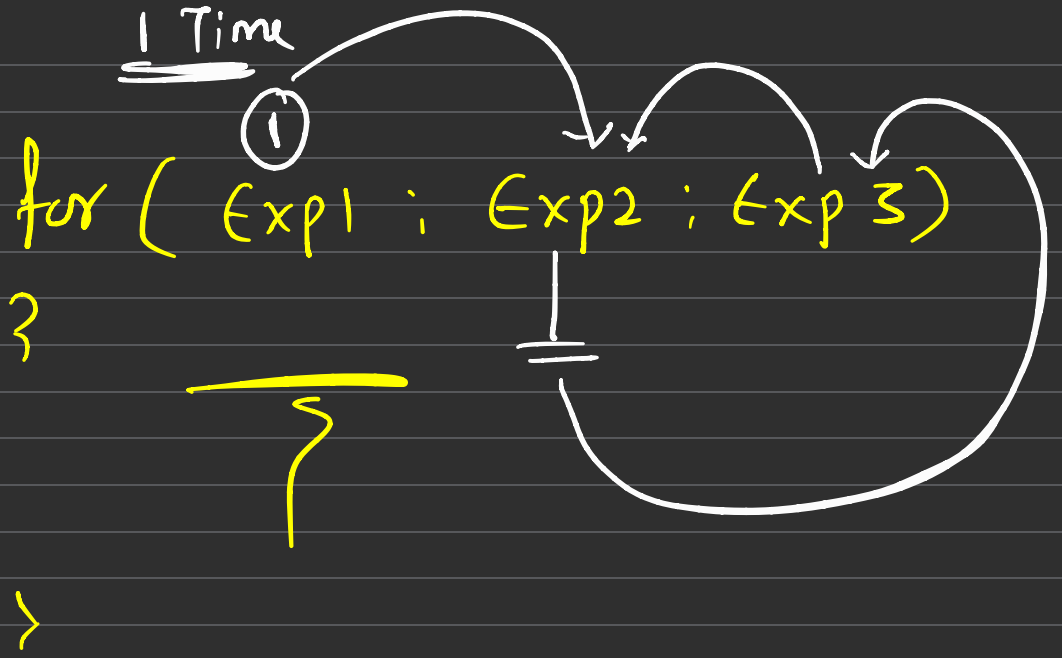
* For

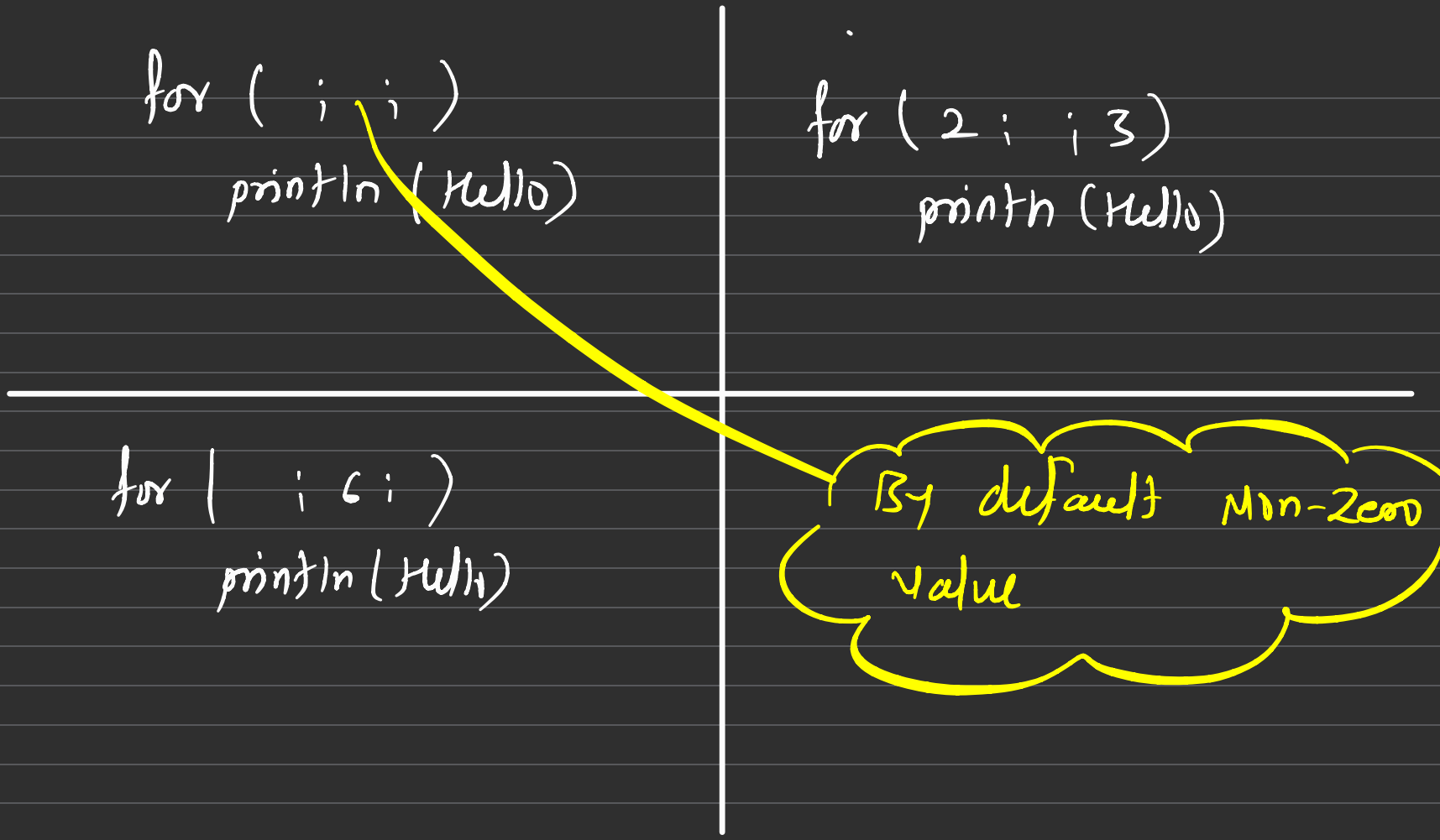
syntax:

for (init ; condition ; inc/dec)

Eg: for (1 ; 2 ; 3)
println ("Hello")

for (Exp1 ; Exp2 ; Exp3)





While

Compulsory

while (Exp / condition)
{

}

inc / dec

first check condition

& then Execute

int x = 12

while (x < 20)
{ printh (Hello)

x++

}

* Do while

```
do
{
}while (condition);
```

```
int x=10
do
{
println(Hello)
x++
}while (x > 100);
```

It first Execute & then check condition