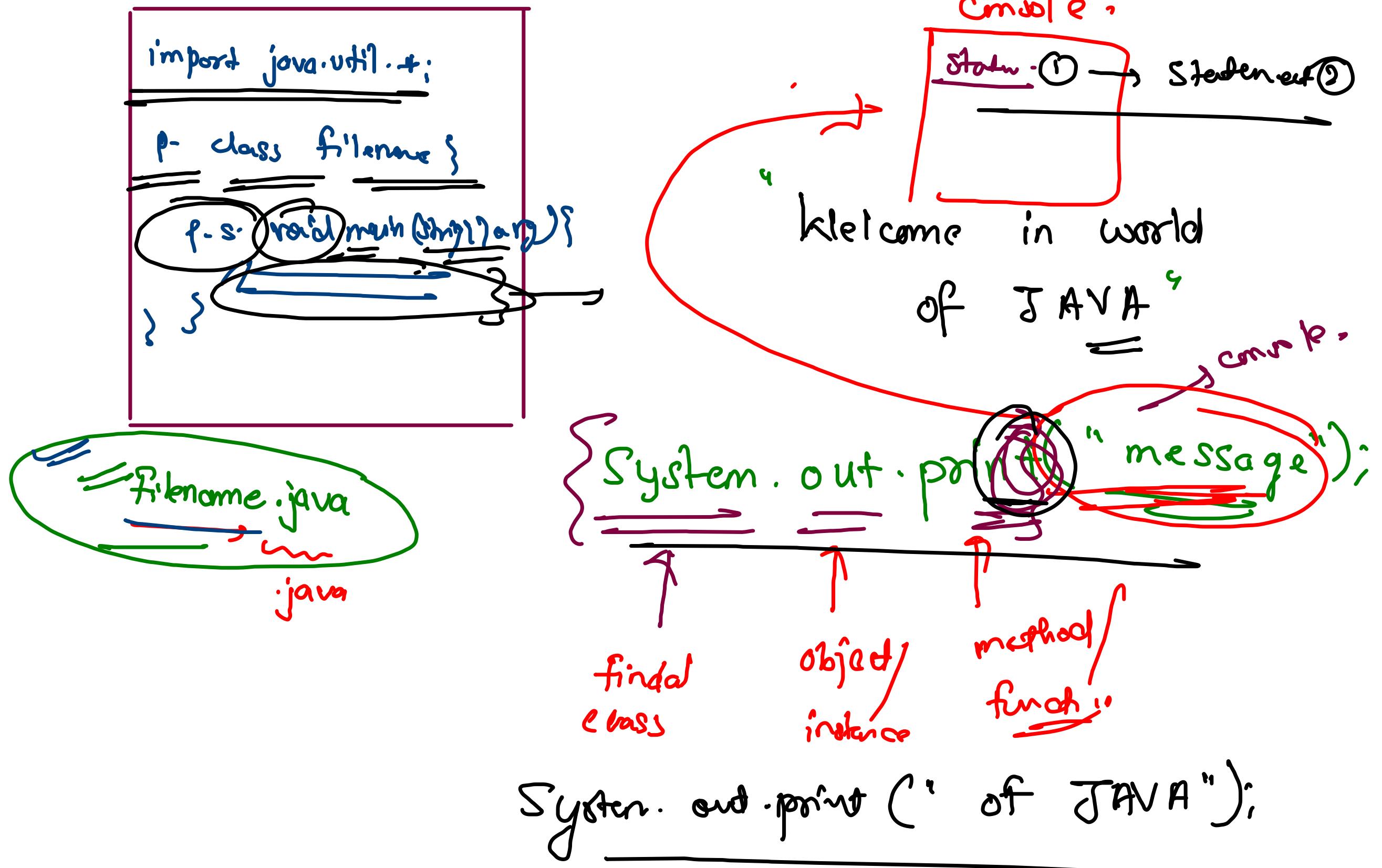
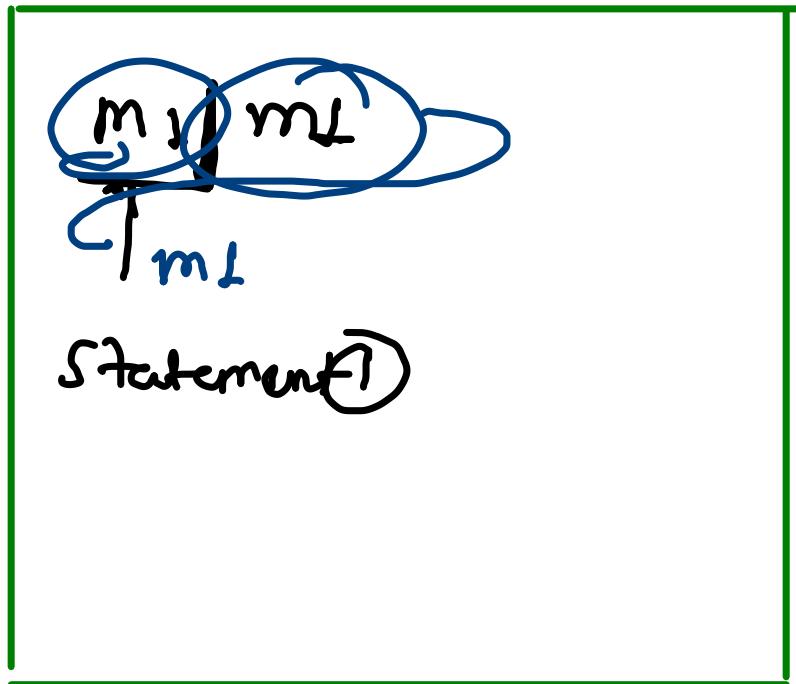


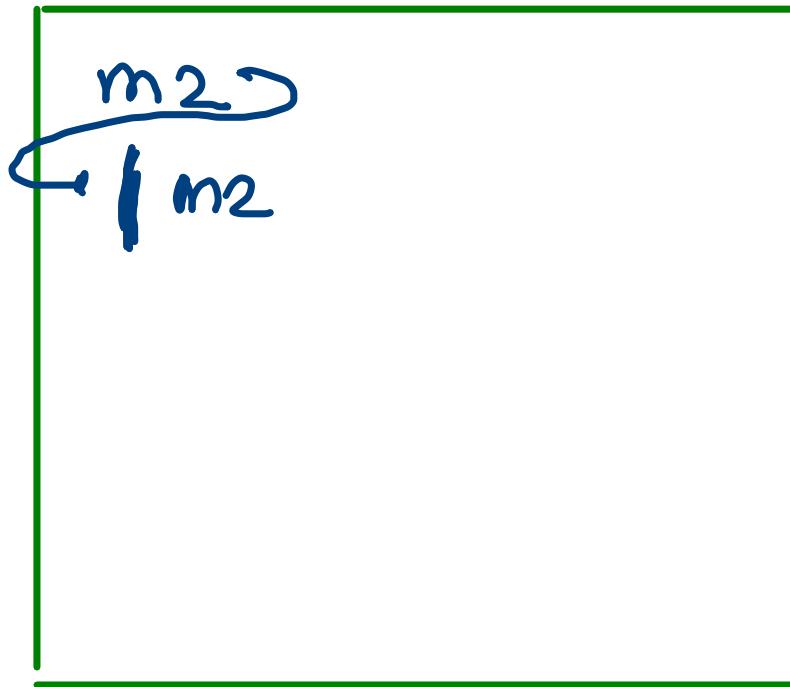
Print →



→ System.out.print(m₁)
→ System.out.print(m₁)
 console

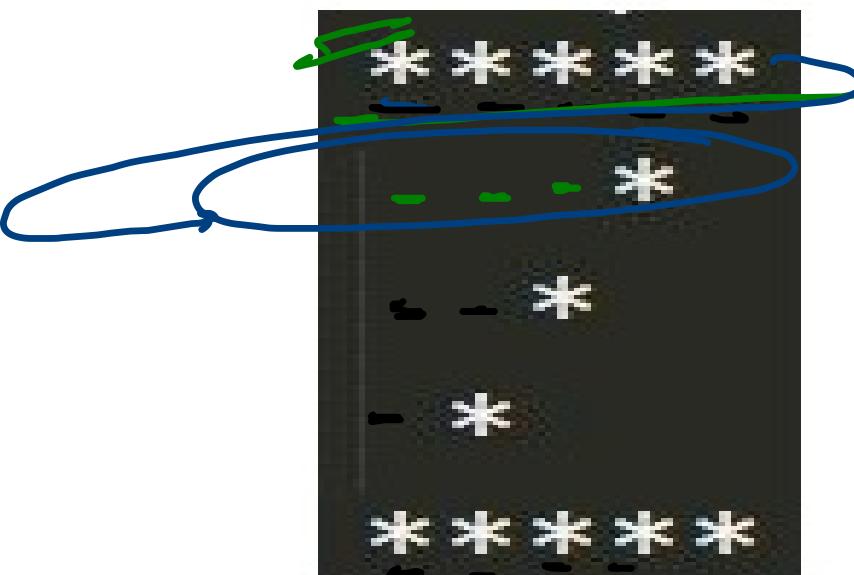


System.out.println(m₂)
System.out.println(m₂)
 console



Point - 2

Hello



* * * * *
- - - *
- - *
- *
+ a a a a a

→ line①

5 Star

→ Shift Enter

→ line②

3 Space

1 Star

→ line③

2 Space

1 Star

→ line④

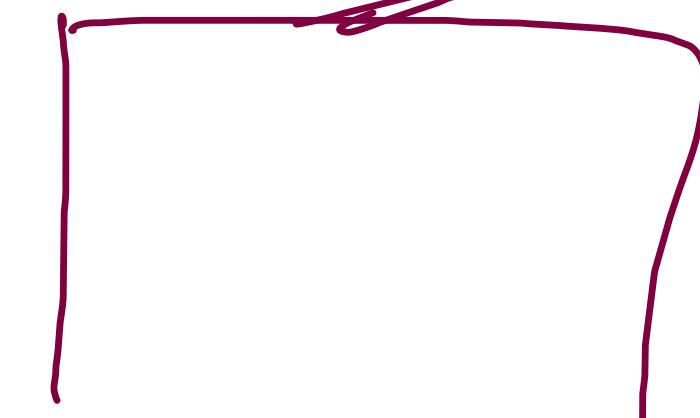
1 Space

1 Star

→ line⑤

5 Star

task



Comments → Beneficial

~~Debugging, (Analysis of code)~~

Single line

para graph
com.

→ Paragraph

// Statement

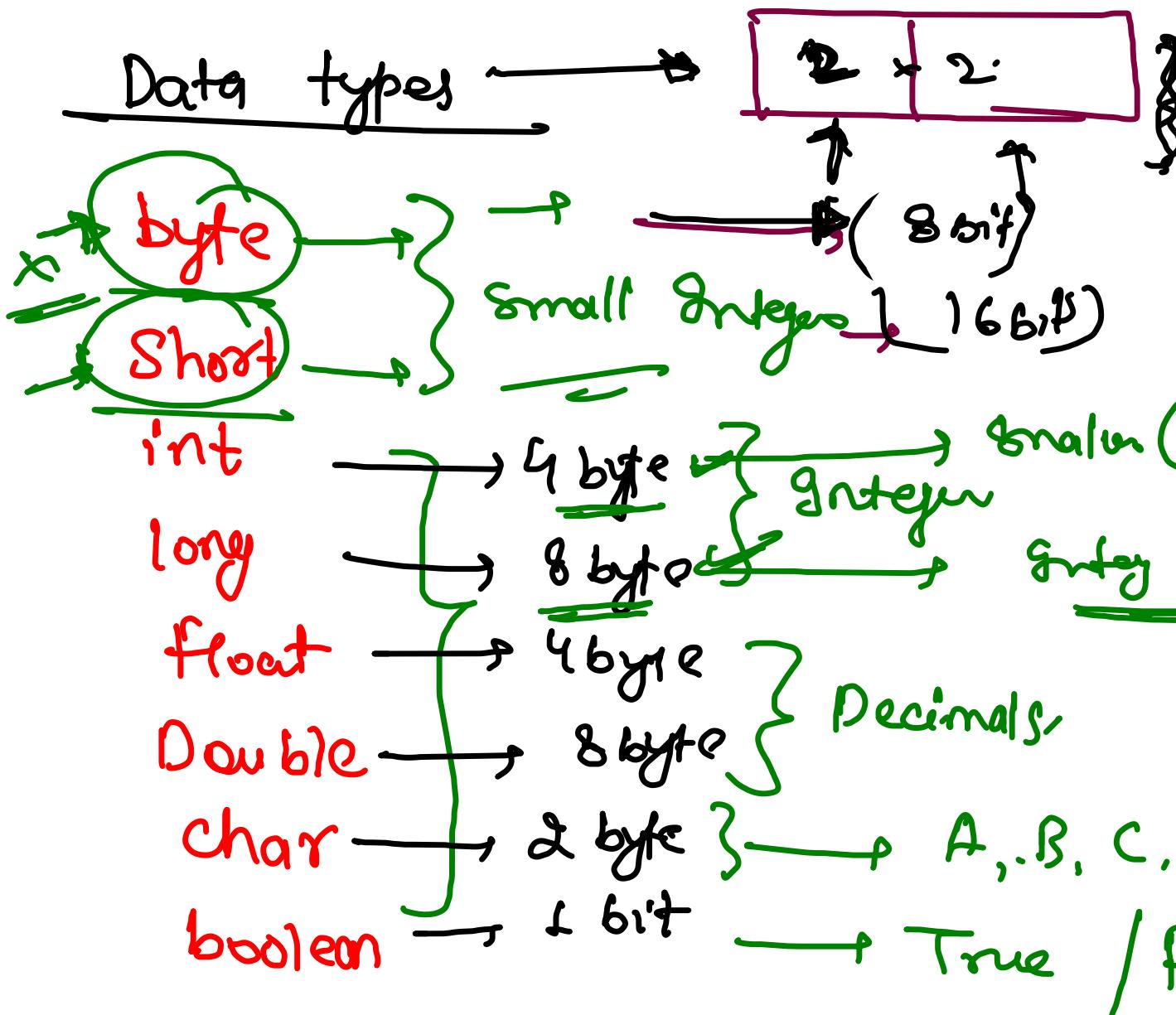
variables

$x = 10;$
 $y = 20;$
 $z = x + y;$
 $z = ? 30$



how to print variables -

Data types →



1 byte = 8 bits -

int $x = 10;$

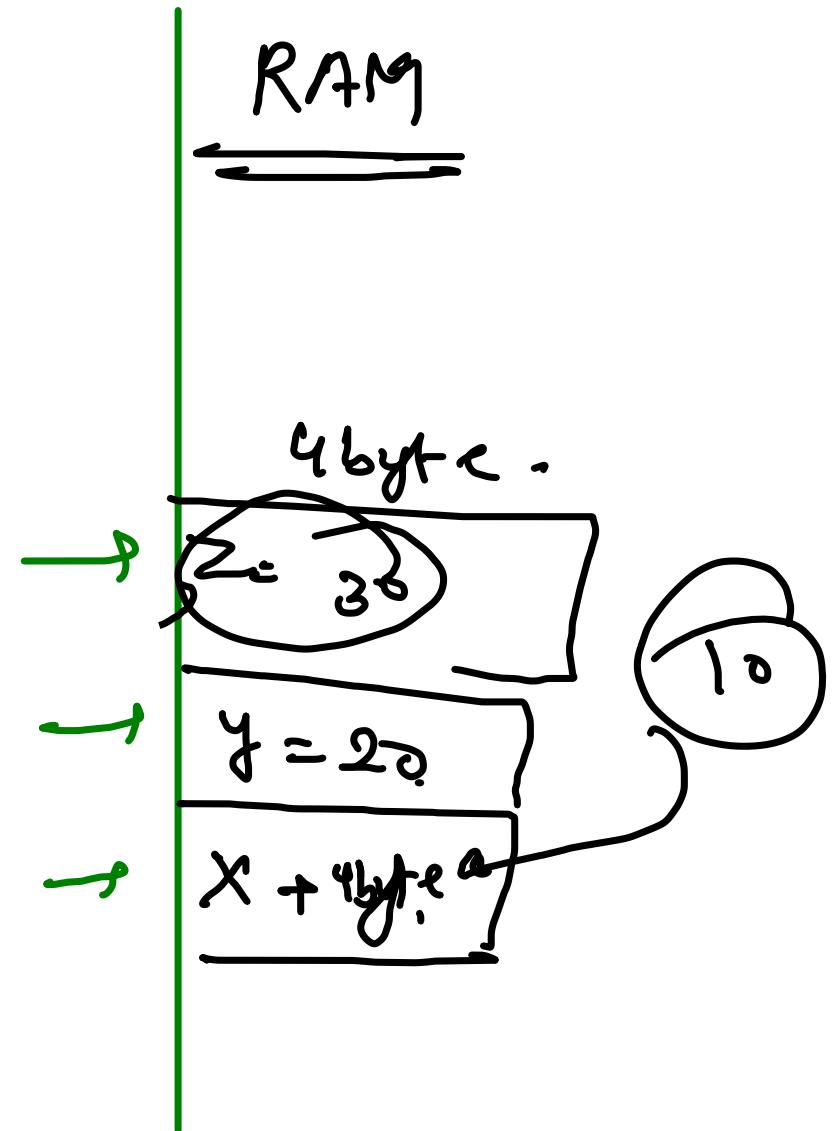
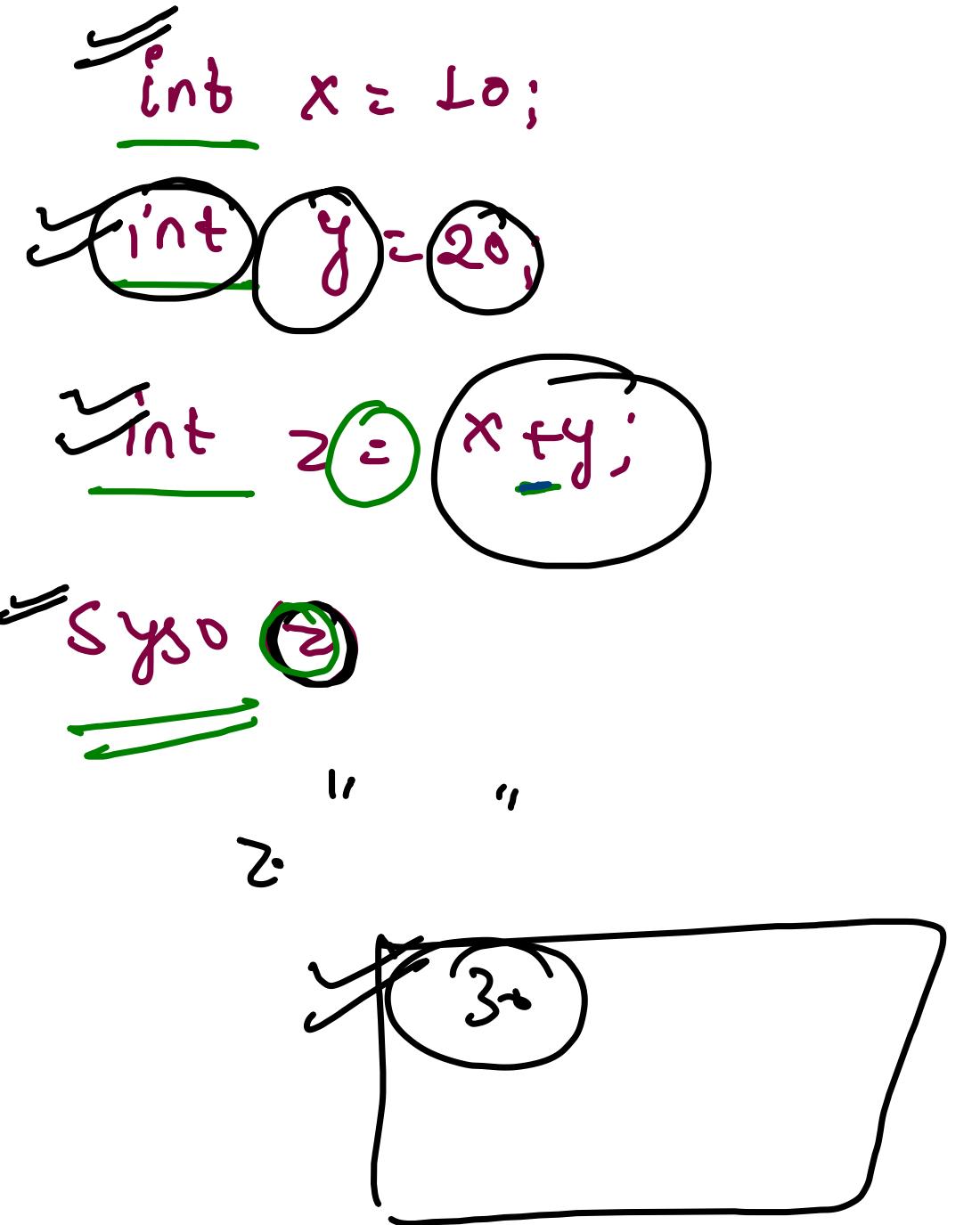
int $y = 20;$

int $z = x + y;$

`System.out.println(z);`

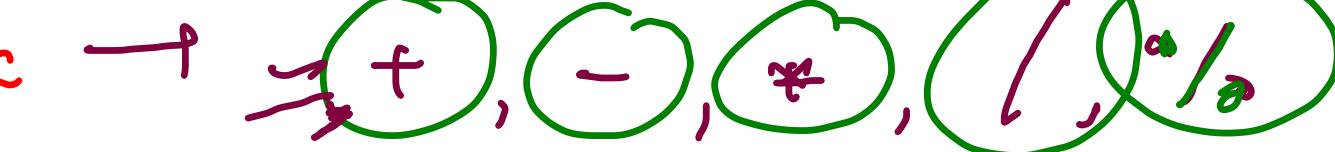
Runtime for long

1, 2, 3, 4,



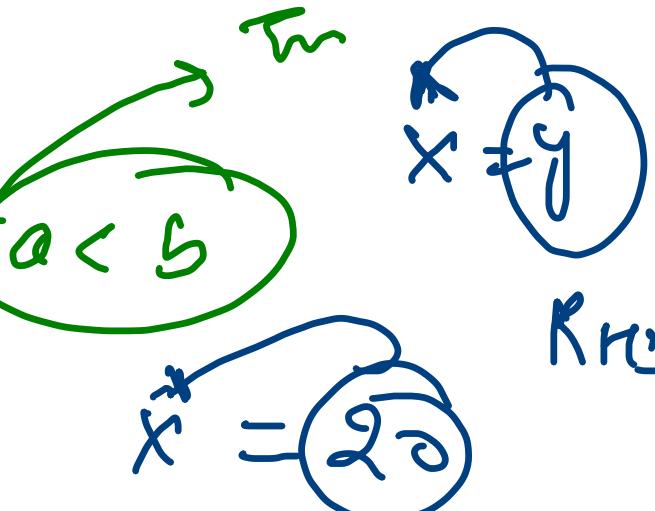
operator \rightarrow

Arithmetic \rightarrow



~~logical~~

$\&$, (AN), $\|$, (OR)



RHS

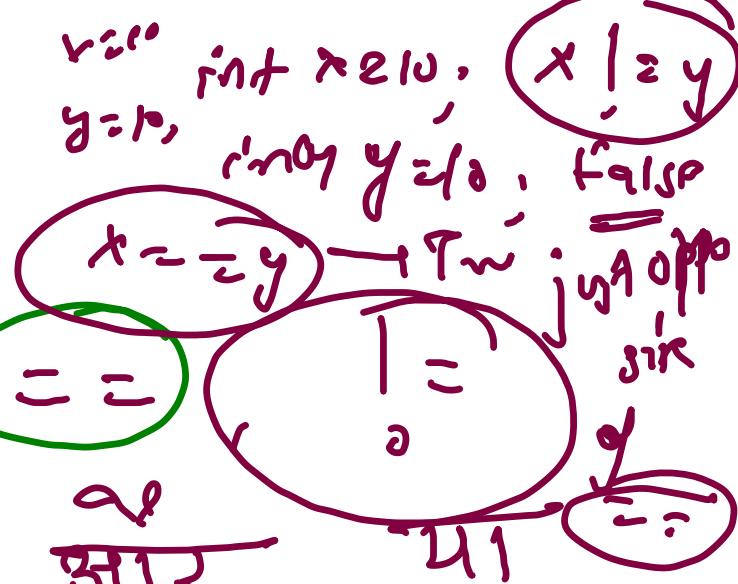
\approx Assignment

<, >, <=, >=

% \rightarrow modulo operator

$$8 \% 2 = 0 \quad 8 \quad \approx \text{Remainder}$$

$$8 \% 3 = 2 \quad 3$$



A	B	A & B	A B
F	F	F	T
F	T	F	T
T	F	F	T
T	T	T	T

```
import java.util.*;
```

```
P class Main {
```

```
    P.S. void main(String[] args) {
```

```
        stat ①
```

```
        stat ②
```

```
        stat ③
```

```
        { stat ④ }
```

```
}
```

```
3
```

```
if true  
cond ②  
if condition ①  
// if - body  
vars are equal  
} else {  
    // else - body  
    if condition is  
    false then else will  
    vars are not executed or  
    ignored.
```

```
if  
x  
else  
x
```

MOTE: Not necessary that if-else
are combined together
single if can exist

if - else
if true
 $x \geq 0, y = 1,$

if -

else -

if true
 $x = y$

if -

else -

if true
 $x = y$

if -

else -

if true
 $x = y$

if -

else -

if true
 $x = y$

if -

else -

if true
 $x = y$

if -

else -

if true
 $x = y$

if -

else -

if true
 $x = y$

if -

else -

if true
 $x = y$

Condition =

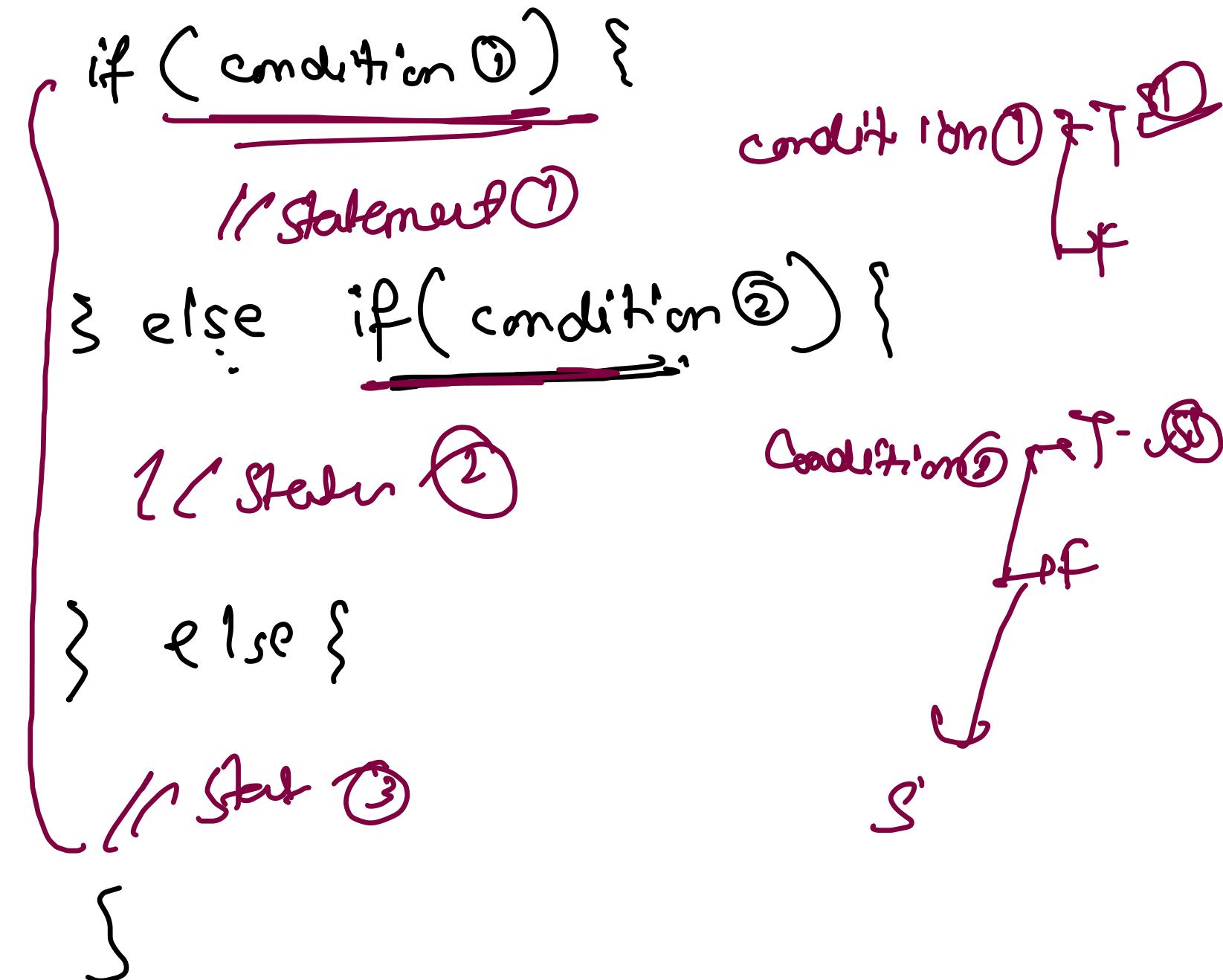
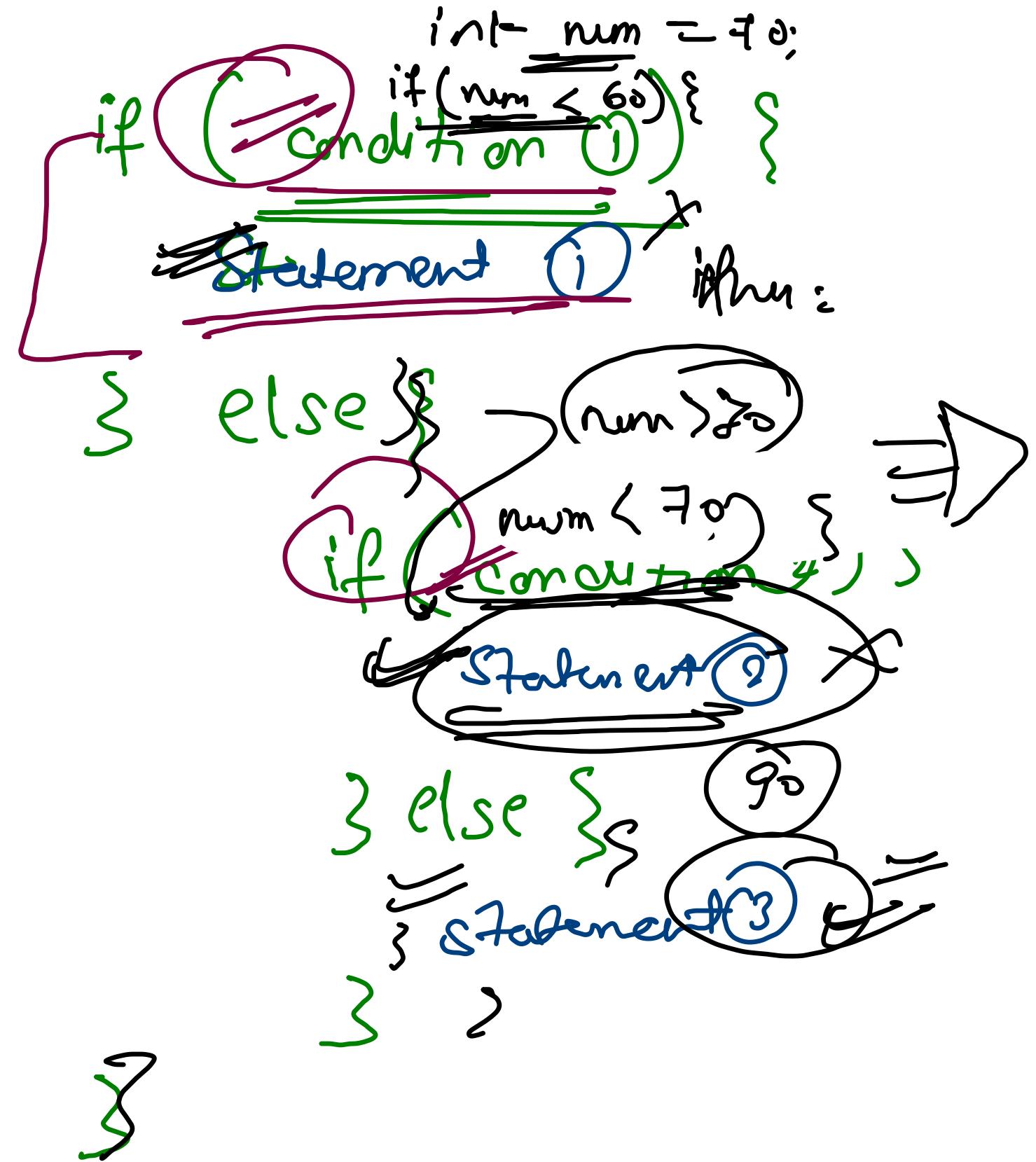
$x = y$

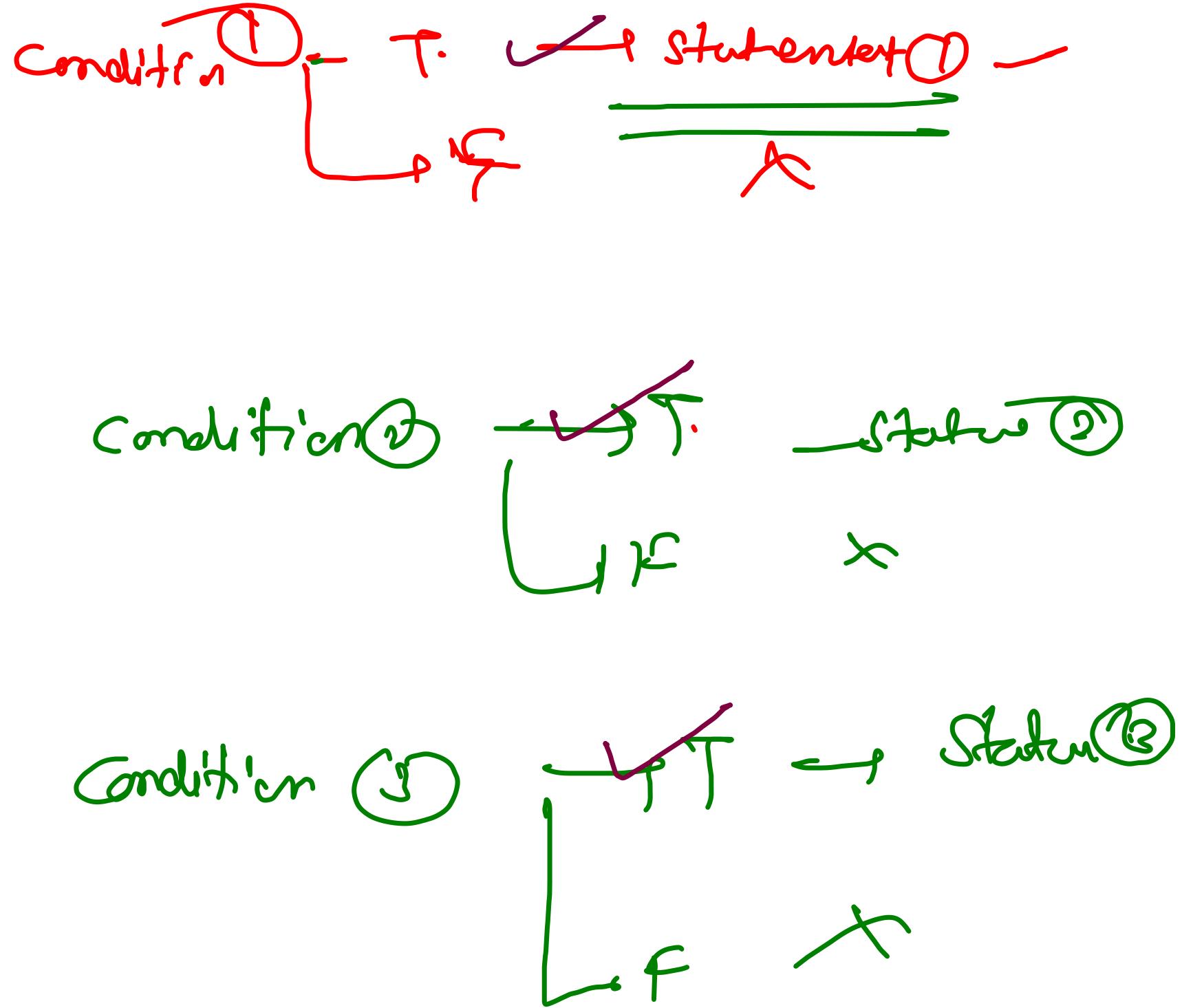
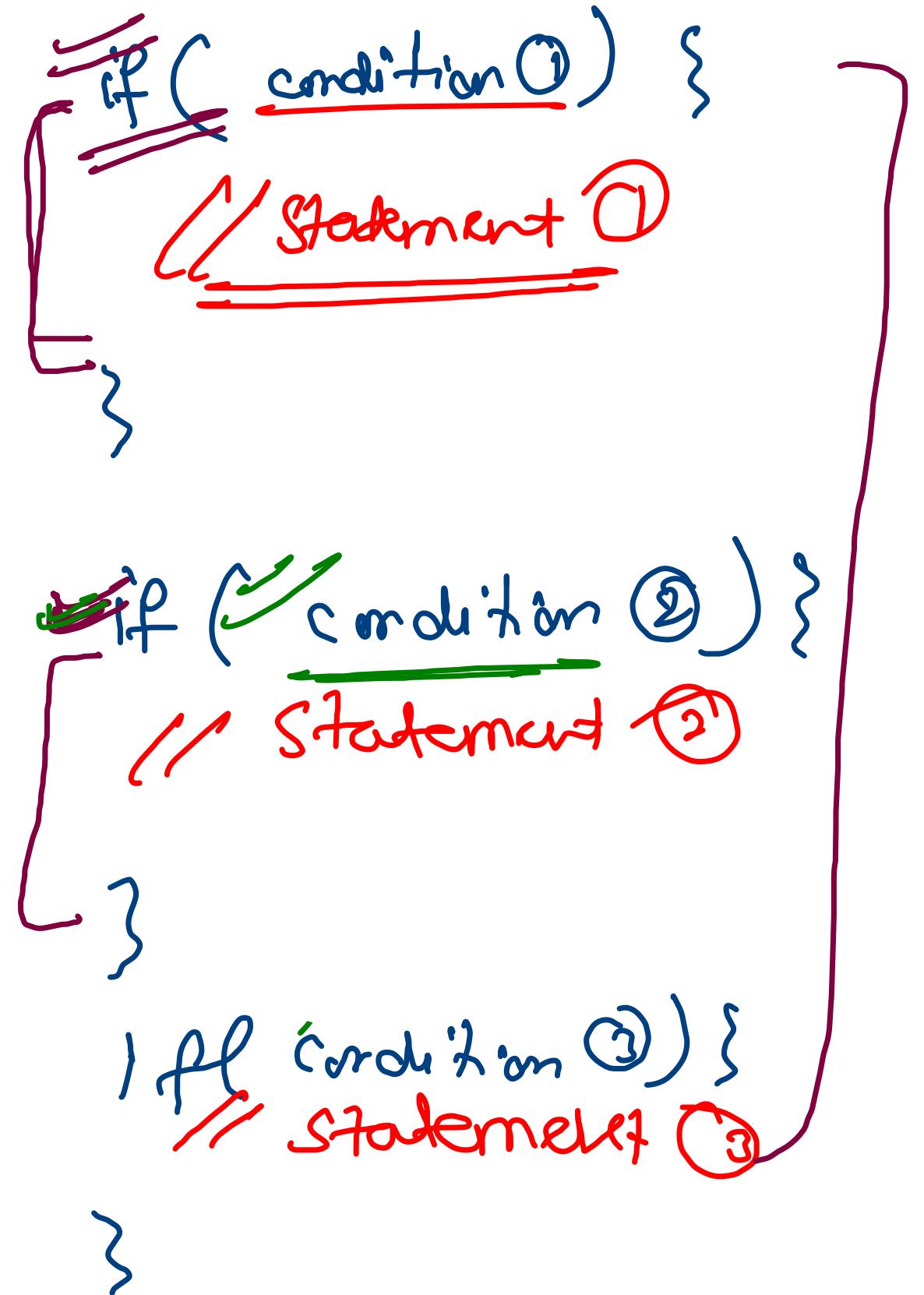
True

if condition is true
the "if - body"
will execute

if condition is
false then else will

Vars are not executed or
ignored.





$a = 100$
 $b = 200$
 $c = 300$

largest of three Number.

if ($a > b \& a > c$) {
 winner $\rightarrow a$ }
 else if ($b > a \& b > c$) {
 b is c finest
 $b > c$ }

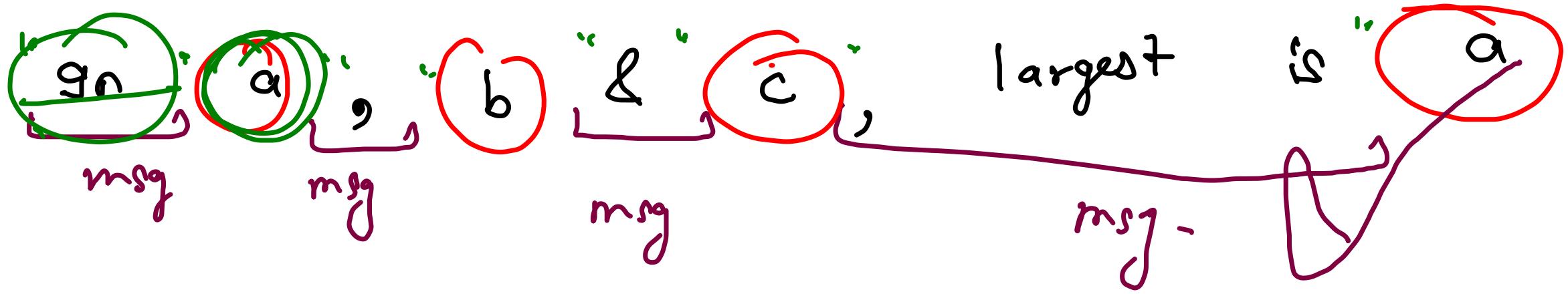
gn $\{ 100, 200, 300 \}$ } else
 largest is $\{ 300 \}$

if ($b > a \& b > c$) {
 $b > c$ }

if a' is a condition {
 j
 exit }

if ($b > a \& b > c$) {
 neg
 neg neg
 neg
 largest } else {
 winner - c }

उत्तर + AND $\rightarrow \&$

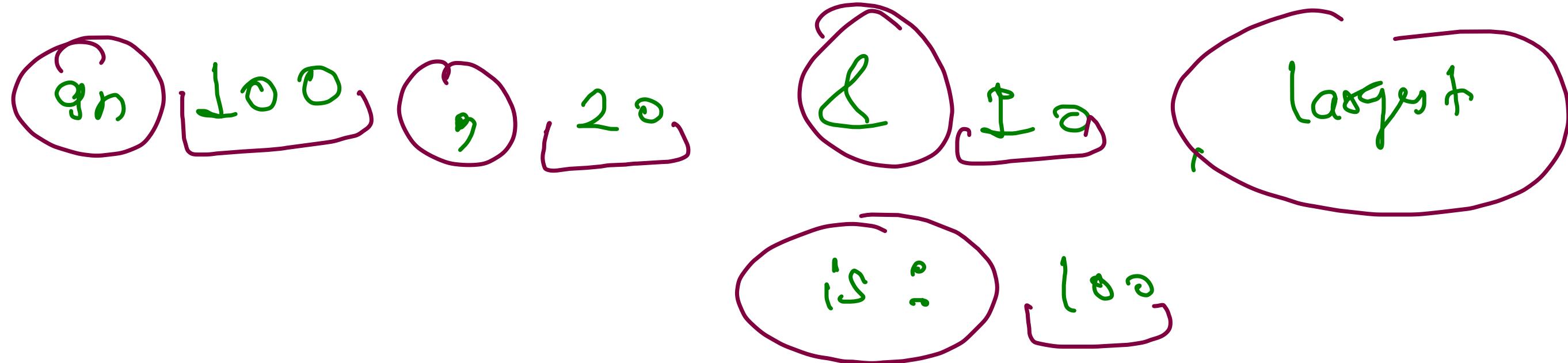


$\varphi \approx 100$

$b \approx 20$

$c = 10$

$$= \cancel{\text{msg}} + \text{Var} + \cancel{\text{msg}} -$$



marks = 95

- ✓ 2.1 for marks above 90, print excellent.
- ✓ 2.2 for marks above 80 and less than equal to 90 print good.
- ✓ 2.3 for marks above 70 and less than equal to 80, print fair.
- ✓ 2.4 for marks above 60 and less than equal to 70, print meets expectations.
- ✓ 2.5 for marks less than equal to 60, print below par.

Hand-writing

```

marks = 95
if( marks > 90 )
    sys( "Excellent" );
else if( marks > 80 )
    sys( "Good" );
else if( marks > 70 ) { && marks <= 80 }
    sys( "Fair" );
else if( marks > 60 ) { && marks <= 70 }
    sys( "Meets Expectations" );
else {
    sys( "Below Par" );
}

```

marks = 78 marks <= 90 marks <= 60

→ condition
expr

obvly

necessary

fair

> 90

no

Smart

```

if( marks > 90 ) {
    sys( "Excellent" );
} else if( marks > 80 ) {
    sys( "Good" );
} else if( marks > 70 ) {
    sys( "Fair" );
} else if( marks > 60 ) {
    sys( "Meets Expectations" );
} else {
    sys( "Below Par" );
}

```

Tool → gnpw + help Tool

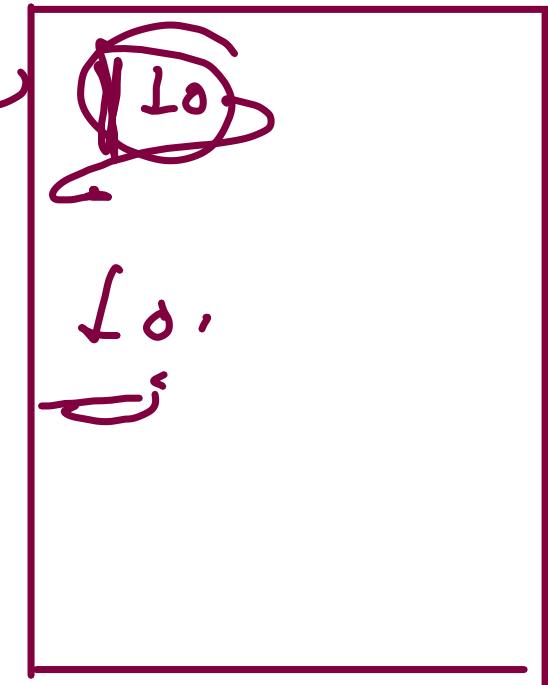
Scanner name of tool (scn) = new Scanner(System.in);

int num = * scn.nextInt();

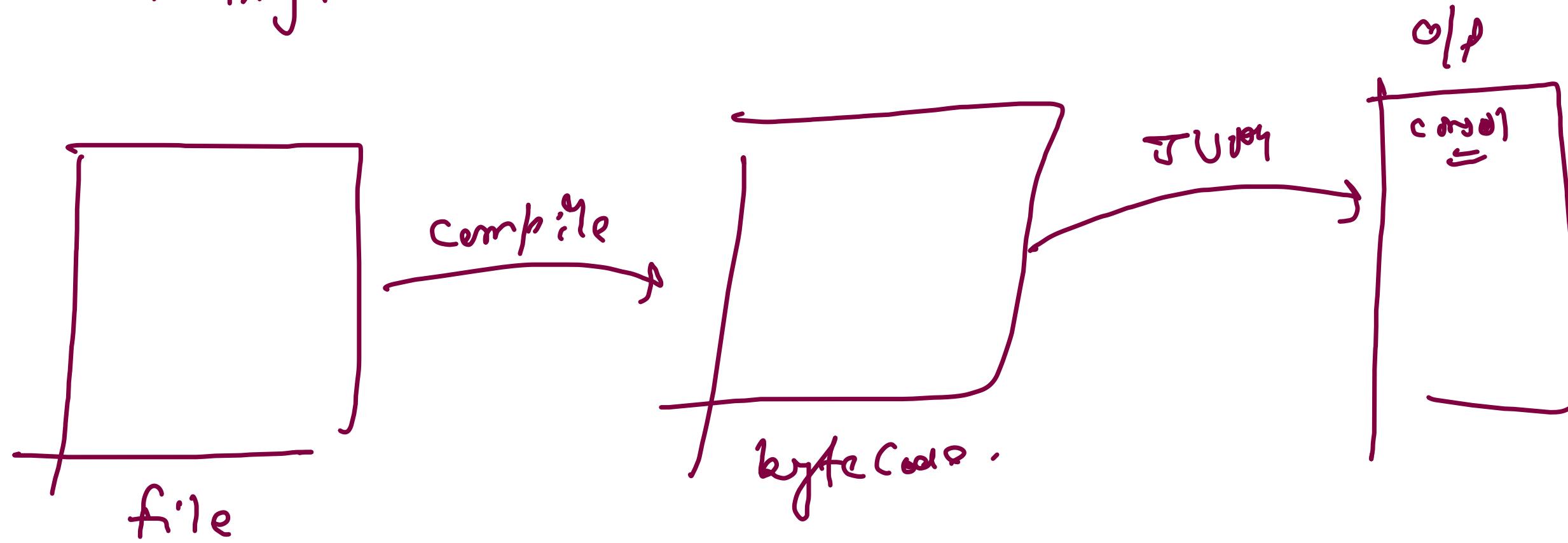
num = 10.

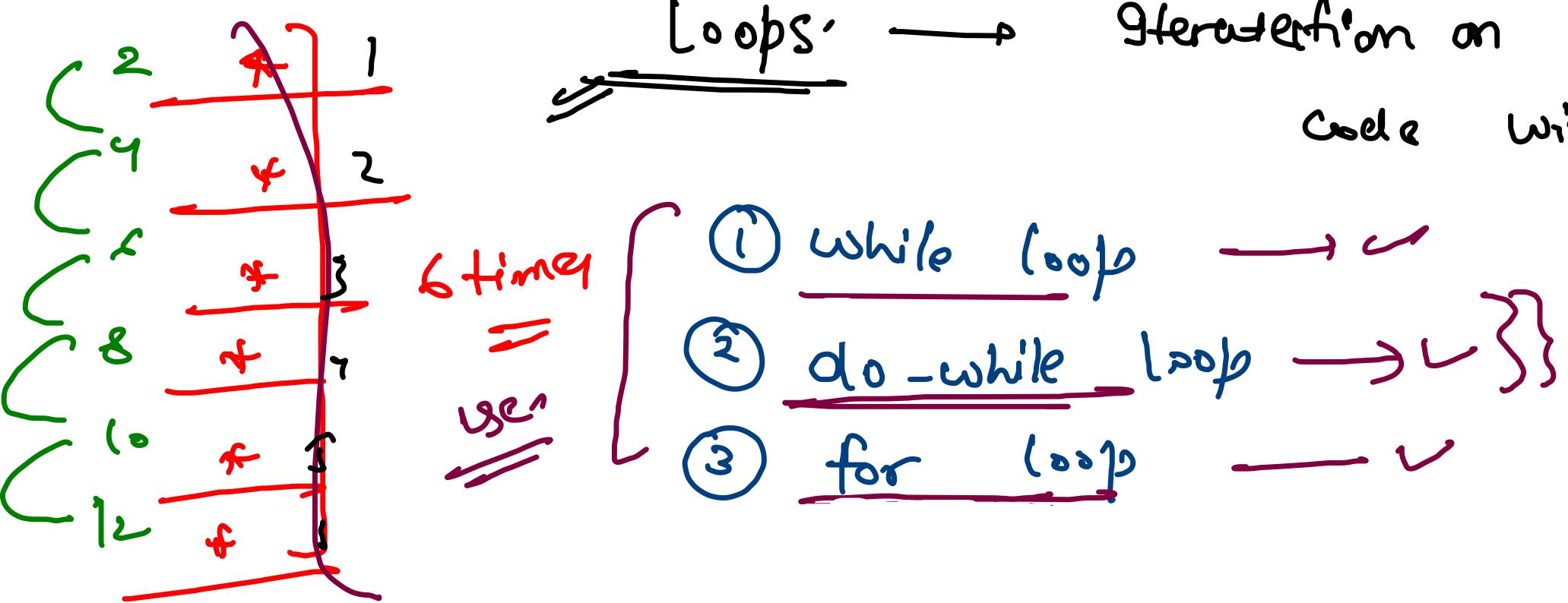
Tool pick next
integer
from console.

System.out.println(num);



javac Main.java





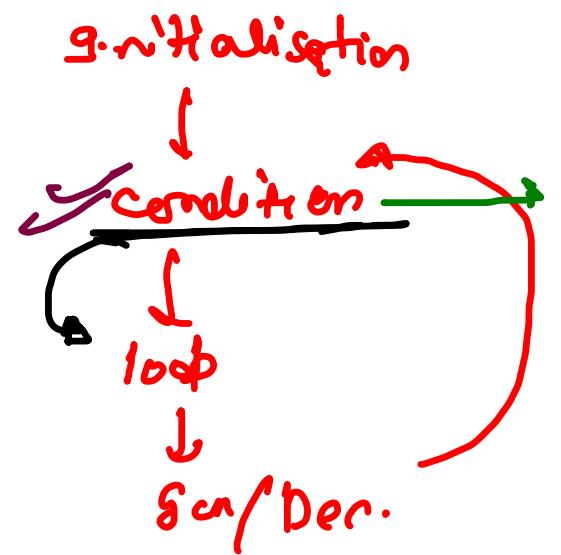
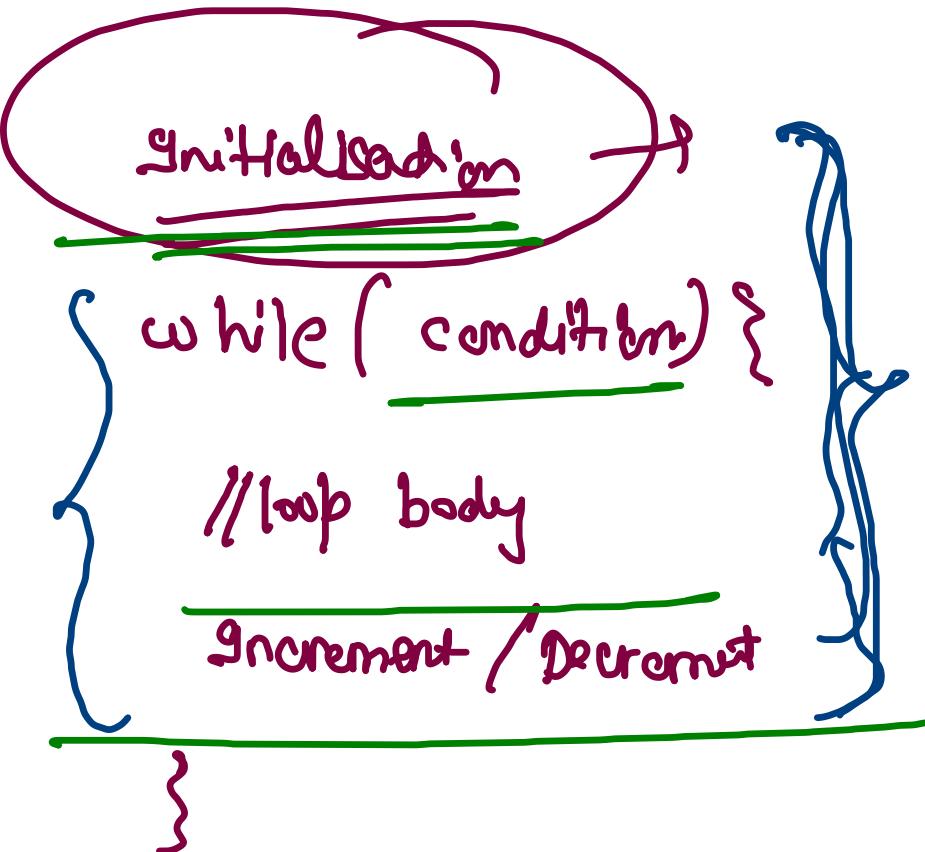
6 times

use

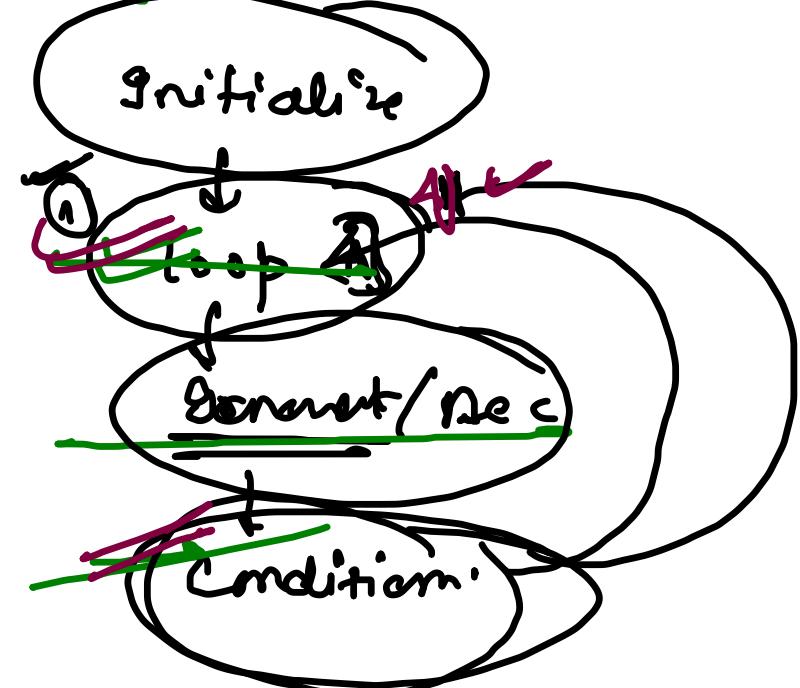
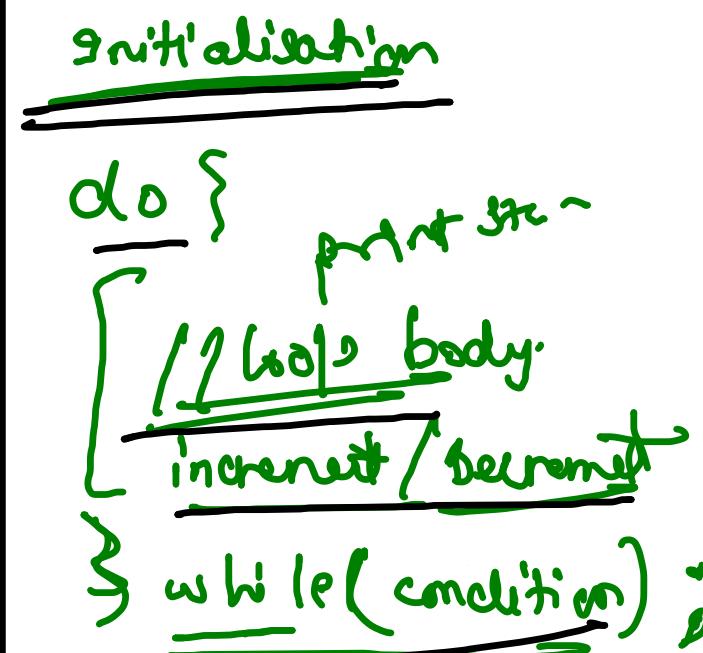
- ① while loop → ↗
- ② do-while loop → ↘ }
- ③ for loop → ↘

Initialisation
Condition →
Increment/Decrement

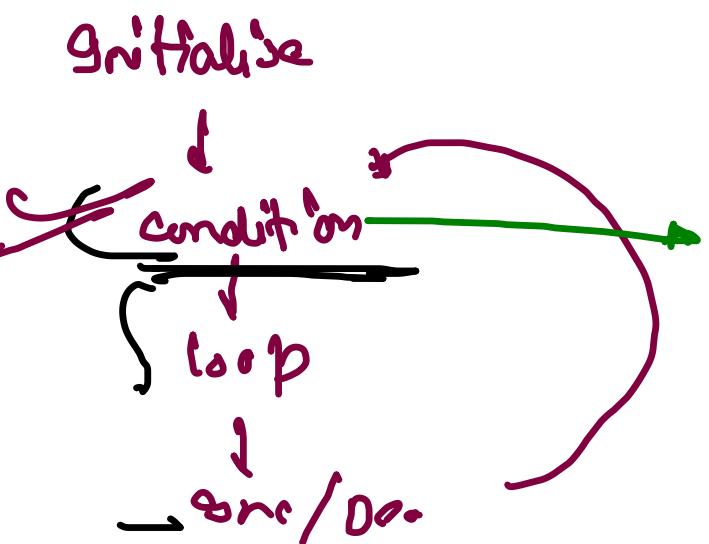
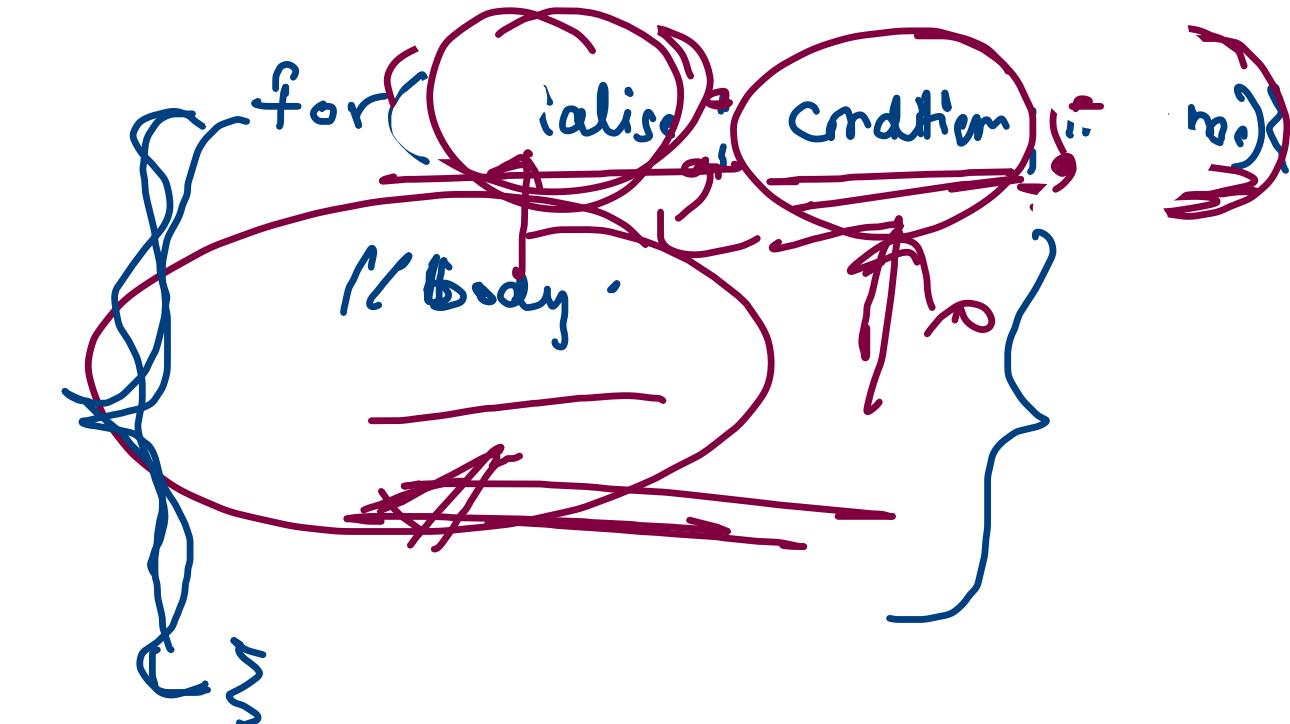
while -loop()



do-while()



for()



~~Node~~ temp = head;

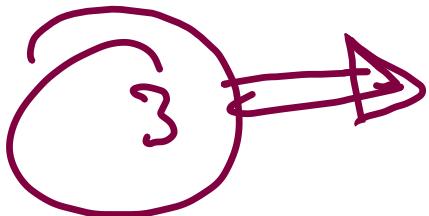
while(temp != null)

 Sys(temp.data);

 temp = temp.next;

}

Schrodw.



for(Node temp = head; temp != null; temp = temp.next)
 Sys(temp.data);

}

M → T → 10:30
T → "
W → "
T → Test
F → Break
S → 2 - 8
S → 2 - 6

→ 2 - getting Str

3 - pattern in
function or array - New variable
number - ↗ D -

any - 2D

Recursion