

## About Me

Incoming SDE Intern + SDE (FTE) @ **Delhivery**  
SDE Intern + SDE (FTE) offer @ **Chaayos**  
Technical Content Writer/Engineer @ **Scaler**  
Technical Content Writer/Engineer @ **PrepBytes**  
Ex – Technical Content Writer @ **InterviewBit**  
Ex – Technical Content Engineer @ **Pepcoding**  
Offers from other Ed-tech institutions like Unstop  
(Dare2Compete), TuteDude, etc.

# Syllabus

① Getting Started

② Arrays & Strings

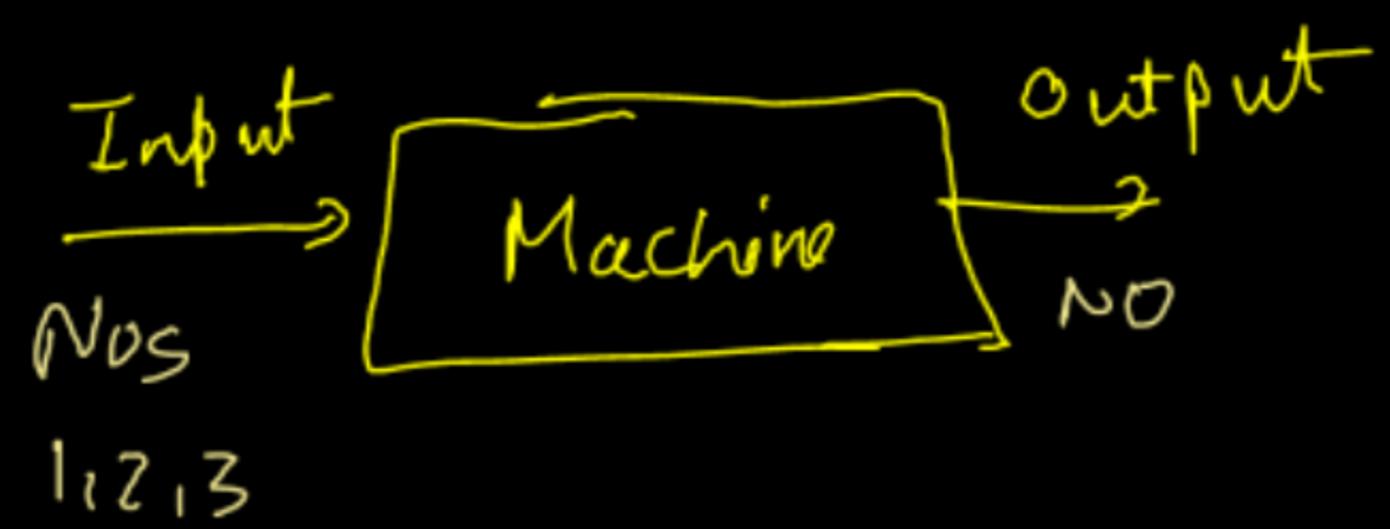
- Two Pointer + Sliding Window
- Greedy Algorithms
- \* Searching & Sorting

## (Data Structures & Algorithms)

- \* ③ linked list
- ⑥ Stack & Queue
- \* ⑦ binary Tree & Binary Search Tree
- ⑧ Hashmap & Heap
- \* ⑨ Graphs
- \* ⑩ Dynamic Programming
- ⑪ Number Theory & Bit Manipulation
- ⑫ Trie

\* Time & Space Complexity

Computer → Human convenience  
→ Machine



## # Communication

Computer understands Binary lang (0 and 1s)

Humans understand Human Lang → instructions  
+ emotions

## Prog Lang



① Computer → Software that converts

Prog Lang to Binary Lang

## Input / output

```
1 // "static void main" must be defined in a public class.  
2 public class Main {  
3     public static void main(String[] args) {  
4  
5     }  
6 }
```

→ comments

```
1 import java.util.*;  
2 public class Main {  
3     public static void main(String[] args) {  
4         //this is a comment  
5     }  
6     //this is also a comment  
7 }
```

## Print in Java

```
1 import java.util.*;  
2 public class Main {  
3     public static void main(String[] args) {  
4         System.out.print("DSA classes");  
5     }  
6 }
```

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         System.out.print("Guneet Malhotra");
5         System.out.print("DSA Classes");
6     }
7 }
```

```
Finished in 69 ms
Guneet MalhotraDSA Classes
```

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         System.out.print("Guneet Malhotra ");
5         System.out.print("DSA Classes");
6     }
7 }
```

```
Finished in 82 ms
Guneet Malhotra DSA Classes
```

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         System.out.print("Guneet Malhotra");
5         System.out.print(" DSA Classes");
6     }
7 }
```

```
Finished in 84 ms
Guneet Malhotra DSA Classes
```

System.out.print('Rohan');

R I      Rohan  
I

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         System.out.println("Guneet Malhotra");
5         System.out.print("DSA Classes");
6     }
7 }
```

```
Finished in 122 ms
Guneet Malhotra
DSA Classes
```

## # Data types in Java

# Type of

containers.

Almirah

Drawer

Fridge



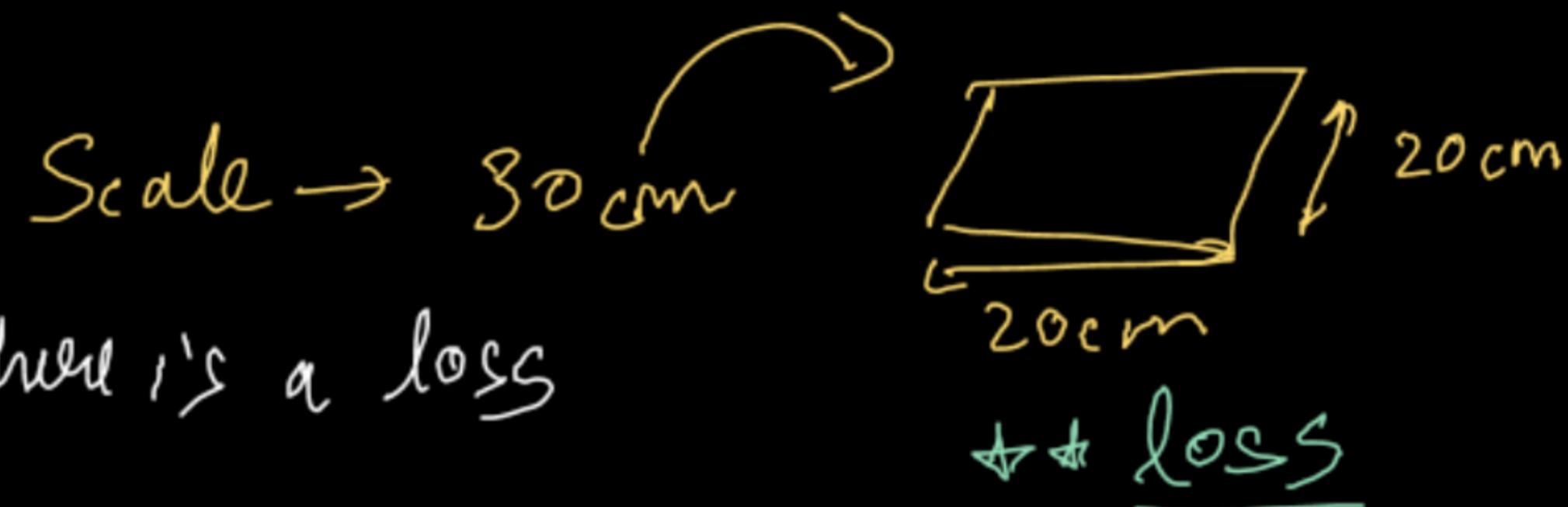
Cloth/clothes



food items

Pen/Pencil

If we try to put item in  
small size container, there is a loss



Size in bytes

Integral (w/o decimal)	{	byte short int long	1 2 4 8	}	30, 50, 700 etc.
---------------------------	---	------------------------------	------------------	---	------------------

floating pt type (with decimal)	{	float double	4 8	}	3.141, 8.632 etc
------------------------------------	---	-----------------	--------	---	------------------

characters  $\rightarrow$  char       $2 \rightarrow A-Z, a-z, 0-9, \#, *, \text{etc.}$

True/false  $\rightarrow$  boolean      Not fixed

11/1

## # Range of Datatype

1 byte = 8 bits

$[-2^{N-1} \text{ to } 2^{N-1} - 1]$   $\rightarrow$  Here N is the no of bits.

## # Range of Short

$$\text{Size} = 2 \text{ bytes} = 2 \times 8 = 16 \text{ bits}$$

$$[-2^{16-1} \text{ to } 2^{16-1} - 1]$$

0 0 0 ... 0  
(16 bits)

$$[-32768 \text{ to } 32767]$$

32768 → 17 bit

## # Data loss

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 100;
5         System.out.println(x);
6     }
7 }
```

Finished in 62 ms  
100

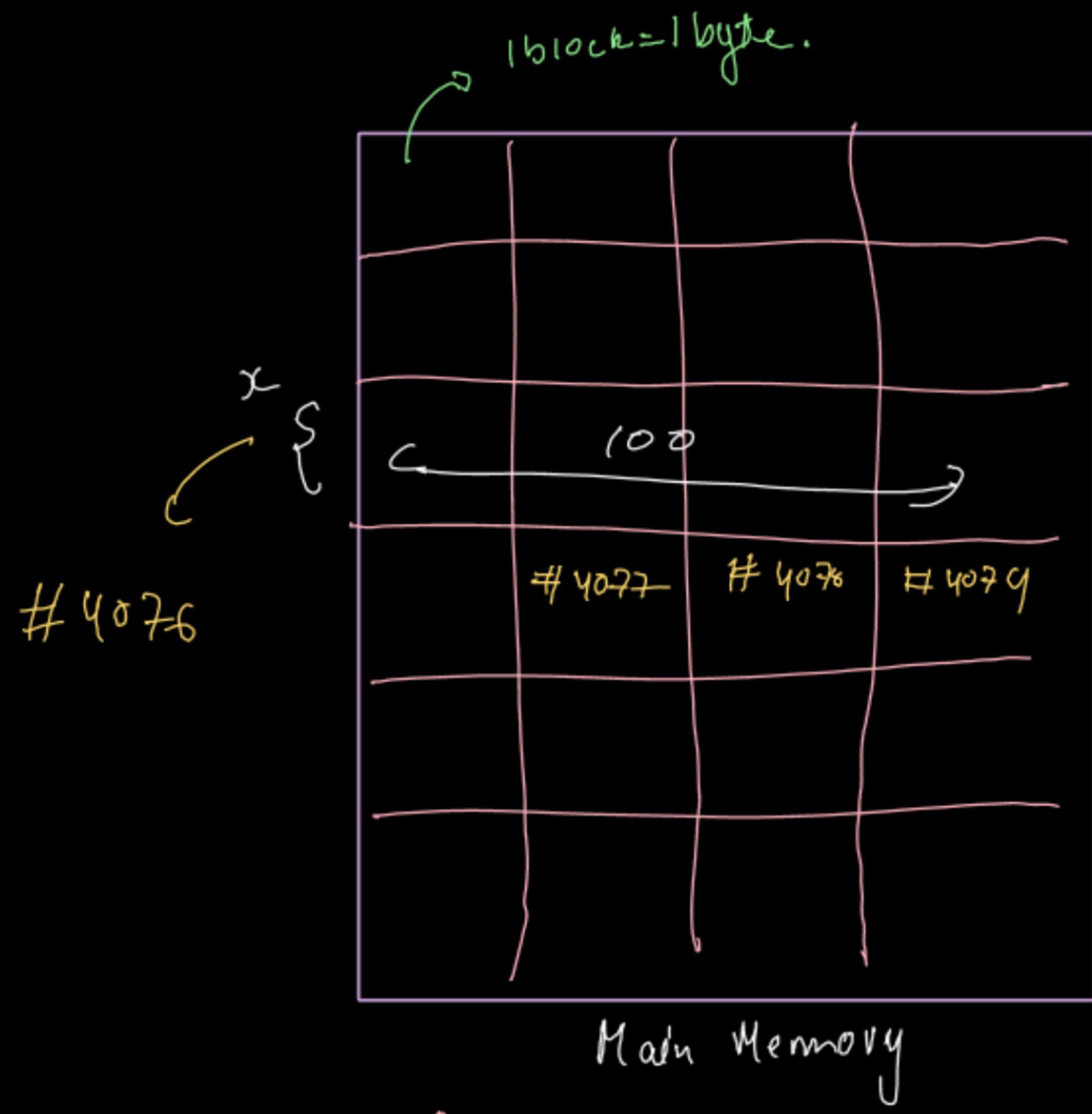
00010 Binary Repres.

32

int x = 100;  
↓              ↗ variable  
datatype      constant

```
int x = 100;
```

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 100; ①
5         System.out.println(x);
6     }
7 }
```



```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 100;
5         System.out.println(x);
6
7         char ch = 'a';
8         System.out.println(ch);
9     }
10 }
```

Finished in 64 ms

```
100
a
```

## # Operators in Java

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 100;
5         System.out.println(x);
6
7         char ch = 'a';
8         System.out.println(ch);
9     }
10 }
```

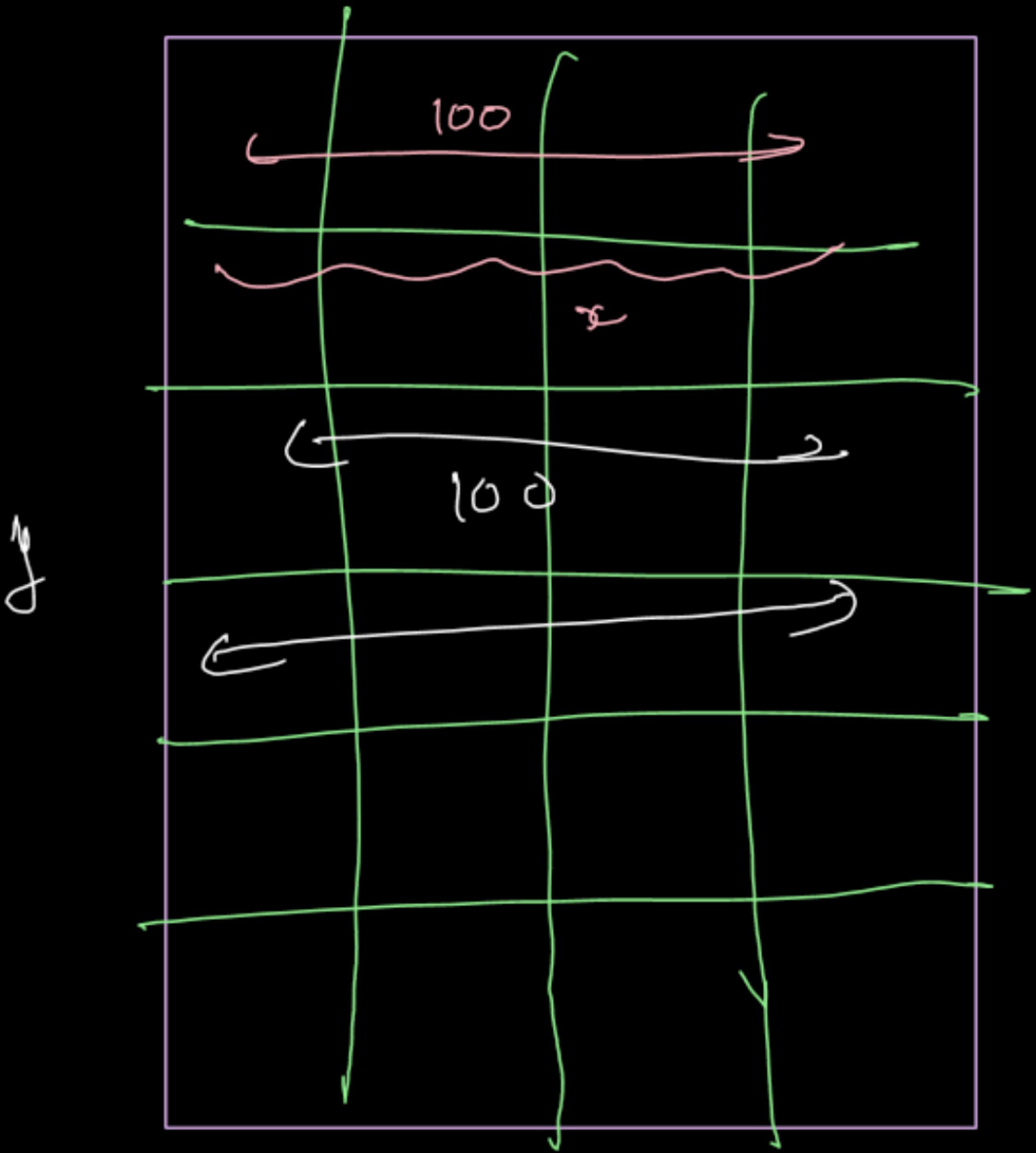
int x = 100;  
  ^    ^  
  LHS   RHS  
assignment operator

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 100;
5         int y = x;
6         System.out.println(x);
7         System.out.println(y);
8     }
9 }
```

# RHS gets assigned to LHS

int y = x;

$\text{int } y = \odot;$



```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x;
5         x = 100;
6         System.out.println(x);
7     }
8 }
```

$$x = \cancel{GIV} 100$$

```
Finished in 50 ms
100
```

\* Initialization of values is a must in Java.

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x;
5         System.out.println(x);
6     }
7 }
```

```
Finished in N/A
Line 5: error: variable x might not have been initialized [in Main.java]
        System.out.println(x);
                           ^
```

# Arithmetic Operators

+

plus

-

minus

\*

multiply

/

divide

%

modulus

int  $x = \underline{1.99}$

= 1

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 10;
5         int y = 30;
6         int res = x / y;
7         System.out.println(res);
8     }
9 }
```

$$10 / 30 = 1 / 3 = 0.\underline{\underline{33}} \quad \times$$

= ⑥

```

1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 10;
5         int y = 30;
6         float res = (float)x / y; //typecasting
7         System.out.println(res);
8     }
9 }
```

→ process of converting INT to another.



integer operator → decimal truncates

# x is converted to float 10 → 10.00

float / int = float

10 / 30 →

$$30 \overline{)10} \quad \begin{matrix} 0 \\ -10 \\ \hline 0 \end{matrix}$$

Quotient is the  
ans of integer / .

10 → remainder is the ans of

modulus .

```

1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         int x = 10;
5         int y = 30;
6         int res = x % y;
7         System.out.println(res);
8     }
9 }
```

Finished in 70 ms

10

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int num = scn.nextInt();
6         System.out.println(num);
7     }
8 }
```

Finished in 108 ms

256

stdin ↴

256

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int num = scn.nextInt();
6         System.out.println(num);
7
8         float f = scn.nextFloat();
9         System.out.println(f);
10    }
11 }
```

Finished in 164 ms

256

10.987

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         int num = scn.nextInt();
6         System.out.println(num);
7     }
8 }
```

```
1 import java.util.*;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner scn = new Scanner(System.in);
5         char ch = scn.nextChar();
6         System.out.println(ch);
7     }
8 }
```

Finished in N/A

```
java.util.InputMismatchException
at line 939, java.base/java.util.Scanner.throwFor
at line 1594, java.base/java.util.Scanner.next
at line 2258, java.base/java.util.Scanner.nextInt
at line 2212, java.base/java.util.Scanner.nextInt
at line 5, Main.main
```

stdin

10.987

Finished in N/A

```
Line 5: error: cannot find symbol [in Main.java]
    char ch = scn.nextChar();
               ^
symbol:   method nextChar()
location: variable scn of type Scanner
```

Next char

is not a  
method .

## # Getting Started (lecture : 2)

→ Conditional statements

- ↳ if - else
- ↳ if - else if - else ladder
- ↳ switch case

} comparison  
operators  
+ logical operators

in Java

(with Memory  
concepts)

→ Loops

- ↳ for
- ↳ while
- ↳ do - while

} continue  
+ break

# Questions

- ↳ Even or odd 1
- ↳ Even or odd 2
- ↳ factorial

## # Conditional Statements

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in); → ① ✓
6         boolean b = scn.nextBoolean(); → ② ✓
7         System.out.println(b); → ③ ✓
8     }
9 }
```

# When we want the code (part of it) to execute in a certain condition.

## If - Else

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         int guessNo = scn.nextInt();
7
8         System.out.println("Start");
9         if(guessNo == 10) { → true
10            System.out.println("You have won the game");
11        } else {
12            System.out.println("You have lost the game");
13        }
14
15        System.out.println("end");
16    }
17 }
18 }
```

```
Finished in 153 ms
Start
You have won the game
end
stdin 10
```

guessNo = 10  
LHS      RHS  
comparison operator

if LHS = RHS  
→ value of this statement is true  
else it is false

$\text{if } (\text{condition}) \{$

$\equiv$        $\xrightarrow{\text{true}}$  if condition is true then this executes

$\} \text{ else } \{$

$\equiv \}$   $\rightarrow$  if condition is false, then this executes

```
1 * import java.util.*;
2 * public class Main {
3
4 *     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         int guessNo = scn.nextInt();
7
8         System.out.println("Start");
9
10        if(guessNo >= 10 && guessNo <= 20) {
11            System.out.println("You have won the game");
12        } else {
13            System.out.println("You have lost the game");
14        }
15
16        System.out.println("end");
17    }
18 }
```

Greater than or equal to

Less than or equal to

fl w

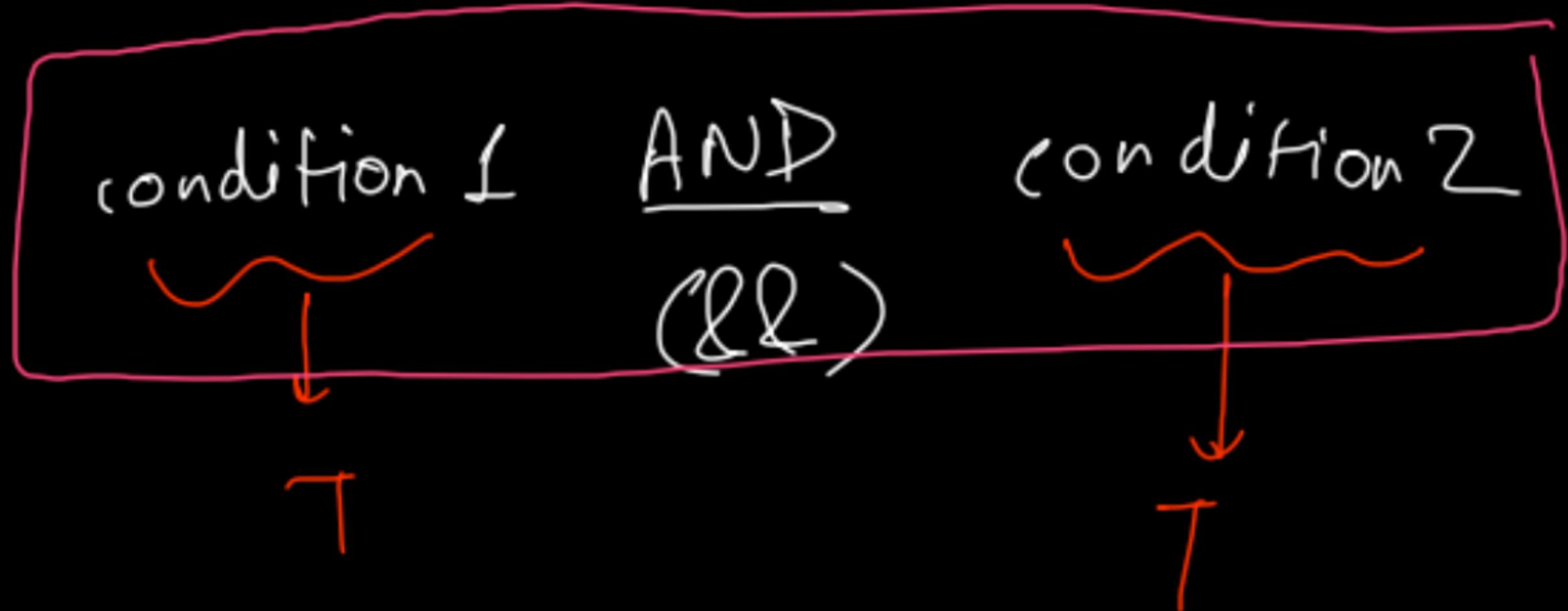
> → greater  
than

< → less than

logical AND



## AND



then AND value will be true  
else value is false

## Logical of

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         int guessNo = scn.nextInt();
7
8         System.out.println("Start");
9
10        if(guessNo == 10 || guessNo == 20) {
11            System.out.println("You have won the game");
12        } else {
13            System.out.println("You have lost the game");
14        }
15
16        System.out.println("end");
17    }
18 }
```

logical OR

cond 1    OR    cond 2



if min one is true  
answer is true

## Logical Not

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         int guessNo = scn.nextInt();
7
8         System.out.println("Start");
9
10        if(! (guessNo == 10 || guessNo == 20)) {
11            System.out.println("You have won the game");
12        } else {
13            System.out.println("You have lost the game");
14        }
15
16        System.out.println("end");
17    }
18 }
```

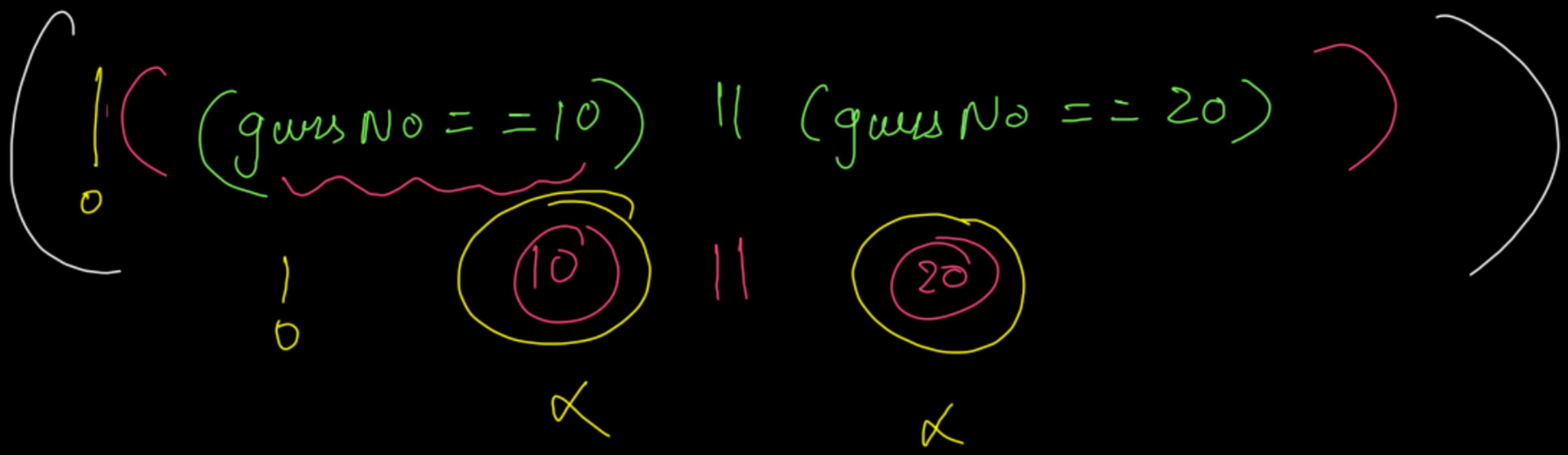
logical not

! (condition)

↳ it negates  
the condition

True → false

false → True



## Not Equal to

```

import java.util.*;
public class Main {

  public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int guessNo = scn.nextInt();

    System.out.println("Start");

    if(guessNo != 10) {
      System.out.println("You have won the game");
    } else {
      System.out.println("You have lost the game");
    }

    System.out.println("end");
  }
}
    
```

Not equal to

LHS is Not equal

to RHS

Then statement  
value is true

else false .

## # If - else if - else ladder

```
import java.util.*;
public class Main {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int guessNo = scn.nextInt();

        System.out.println("Start");

        if(guessNo == 10) {
            System.out.println("You have won the game");
        } else if(guessNo == 20) {
            System.out.println("You have won the game");
        } else {
            System.out.println("You have lost the game");
        }

        System.out.println("end");
    }
}
```

if (condn1) {  
    }  
    }  
} else if (condn2) {  
    }  
    }  
    }  
    }  
    } only this  
    } will execute  
    }  
    } else {  
        }  
    }

## Multiple Else if

```
import java.util.*;
public class Main {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int guessNo = scn.nextInt();

        System.out.println("Start");

        if(guessNo == 10) {
            System.out.println("You have won the game");
        } else if(guessNo == 20) {
            System.out.println("You have won the game");
        } else if(guessNo == 30) {
            System.out.println("You have won the game");
        }
        else {
            System.out.println("You have lost the game");
        }

        System.out.println("end");
    }
}
```

```
if (condition) {  
    } ==>  
    if ( )  
    {  
        }  
    } ==>  
    if (condn1) {  
        } ==>  
        .  
        else if (condn2) {  
            } ==>  
            if (condition3) {  
                } ==>  
                ✓
```

# If there are multiple if statements one after another  
then they are checked.

```
1+ import java.util.*;
2+ public class Main {
3
4+     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         int x = scn.nextInt();
7         int y = scn.nextInt();
8
9         System.out.println("Start");
10
11+        if(x == 10) {
12            System.out.println("You have won the game");
13        } else if(x == 20) {
14            System.out.println("You have lost the game");
15        }
16+        if(x == 100) {
17            System.out.println("You have won the game");
18        }
19
20         System.out.println("end");
21     }
22 }
```

```
Finished in 163 ms
Start
You have lost the game
end
stdin
20
100
```

# for loop → Print your name 10 times

```
• import java.util.*;
• public class Main {
    •     public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        for(int i=1;i<=10;i++) {
            System.out.println("DSA Classes");
        }
    }
}
```

Finished in 110 ms  
DSA Classes  
DSA Classes

A Generally used when exact no of iterations is known.

initial condition

for (int  $i = 1$ ;  $i \leq 10$ ;  $i++$ ) {

Syso (" DSA "); ②

}

$i++ \rightarrow i = i + 1$

(1) + 1

↑  
(2)

DSA }      DSA }

DSA      DSA

DSA  
DSA  
;

$\boxed{xyz}$  3 4 ... 10  
i  
11

## Counting 1 to N

```
✓ import java.util.*;
✓ public class Main {

✓     public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);

        int num = scn.nextInt();

        for(int i=1;i<=num;i++) {
            System.out.println(i);
        }
    }
}
```

Sum of 1 to N

num = 10

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6
7         int num = scn.nextInt();
8         int sum = 0; ① ③
9         for(int i=1;i<=num;i++) {
10             sum = sum + i; ②
11         }
12
13         System.out.println(sum);
14     }
15 }
```

~~X<sub>2</sub>~~

i

~~X<sub>1</sub>~~

Sum

$$\text{Sum} = \text{Sum} + i$$

$$(1) + (2)$$

$$= ②$$



```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6
7         int num = scn.nextInt();
8         int sum = 0;
9         for(int i=1;i<=num;i++) {
10             sum += i; //sum = sum + i
11         }
12
13         System.out.println(sum);
14     }
15 }
```

## # While Loop

(When we don't know exact no  
of iterations. But, we know the condition)

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6
7         int i = 1; ↳ initialization
8     ① while(i <= 10){ ↳ condition
9         System.out.println("DSA Classes"); ↳ ②
10        i++; ③
11    }
12 } ↳
```

12 3..11  
o  
l

DSA

DSA

.

DSA

## Count N to 1

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        while(n >= 1) { ①
            System.out.println(n); ②
            n--; ③
        }
    }
}
```

$$n-- \rightarrow n = n - 1$$

~~10 9~~ 8 7 6 5 4 3 2 1

~

10  
9  
8  
7  
6  
5  
4  
3  
2  
1

1

## # Do while loop

```
import java.util.*;  
public class Main {  
  
    public static void main(String[] args) {  
        Scanner scn = new Scanner(System.in);  
  
        int i = 1; → initialization  
        do{  
            System.out.println("DSA"); → ①  
            i++; → ②  
        } while(i<=10); → ③  
    }  
}
```

1 ↗ 3 ↗ 4

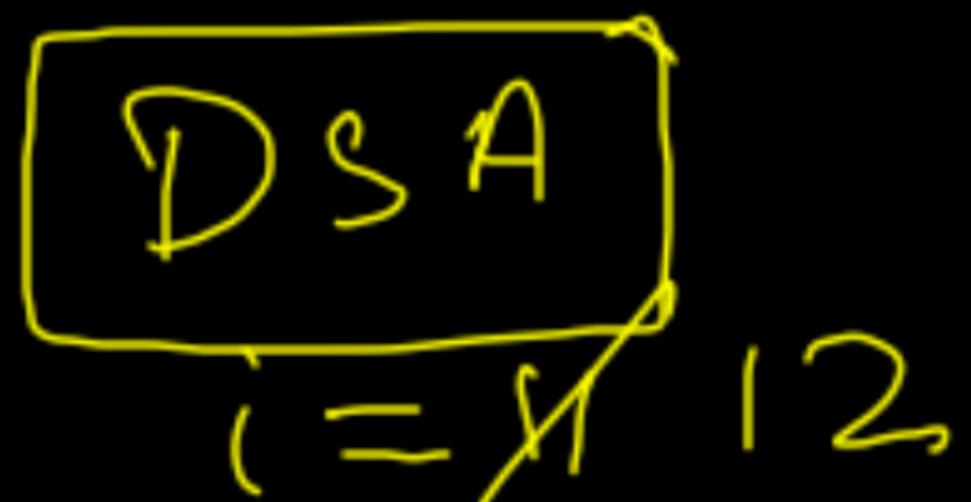
DS A ↗  
DS A  
DS A  
⋮  
DS A ↘

```
int i = 11;    Ⓛ  
while(i <= 10) {  
    System.out.println("DSA");  
    i++;  
}
```

$$11 \leq 10 \rightarrow \text{false}$$

↗ Loop not executed

```
int i = 11;  
do{  
    System.out.println("DSA"); Ⓛ  
    i++;  
} while(i<=10);
```



$$12 \leq 10 \propto$$

# Continue in loops

42 BY

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         for(int i=1;i<=10;i++) {
7             if(i % 3 == 0) {
8                 continue; ①
9             }
10            System.out.println(i); ②
11        }
12    }
13 }
```

```
Finished in 119 ms
1
2
4
5
7
8
10
```

1

2

4

$i \% 3 == 0$

↓ Skips the current iteration

## Break in loops

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6
7         for(int i=1;i<=10;i++) {
8             if(i % 3 == 0) {
9                 break; _____
10            }
11            System.out.println(i);
12        }
13    }
14 }
```

→ break from the loop  
i.e exit the loop.

# # Switch Case

```
1 import java.util.*;
2 public class Main {
3
4     public static void main(String[] args) {
5         Scanner scn = new Scanner(System.in);
6         int num = scn.nextInt();
7         switch(num){ → or character or String
8             case 10:
9                 System.out.println("Option 10");
10                break;
11            case 20:
12                System.out.println("Option 20");
13                break;
14            case 30:
15                System.out.println("Option 30");
16                break;
17            default:
18                System.out.println("Please select out of 10, 20 or 30");
19        }
20    }
21}
```