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

Time & Space Complexity

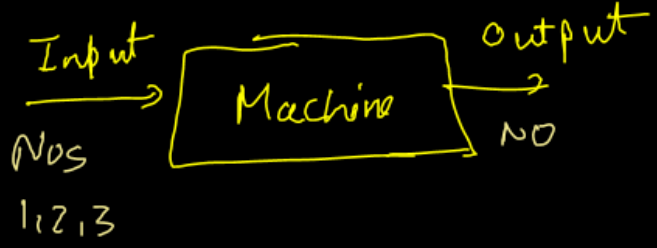
1. Getting Started
2. Arrays & Strings
   - → Two Pointer + Sliding Window
   - → Greedy Algorithms
   - → ☆ Searching & Sorting
3. ☆ Recursion & Backtracking
4. Object Oriented Programming (Java)

Data Structures & Algorithms

5. 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

Computer → Human convinience

↳ Machine



Communication

Computer understands Binary Lang (0 and 1s)

Humans understand Human Lang → instructions

Prog Lang

① Compiler → Software that converts Prog Lang to Binary Lang

Input / output

```java
// "static void main" must be defined in a public class.
public class Main {
    public static void main(String[] args) {

    }
}
```

comments

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        //this is a comment
    }
    //this is also a comment
}
```

Print in Java

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

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

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

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

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

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

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

System.out.print("Rohan");

RI

Rohan
I

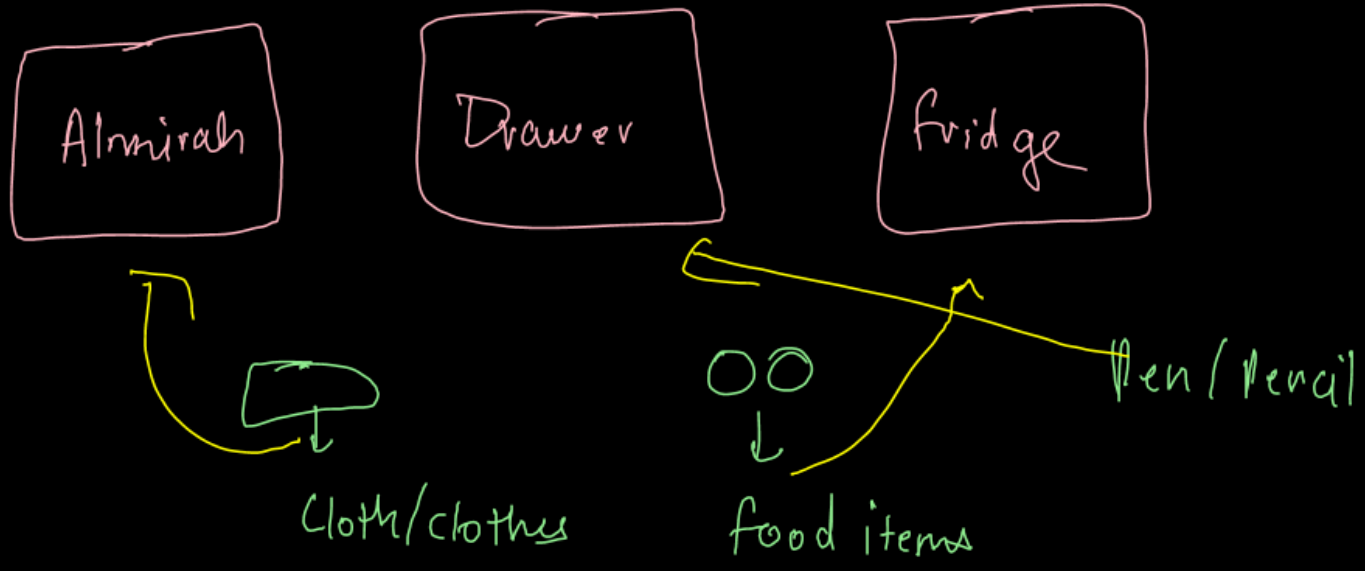
```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        System.out.println("Guneet Malhotra");
        System.out.print("DSA Classes");
    }
}
```

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

stdin

Data types in Java

Type of container.



If we try to put item in

		Size in bytes	
Integral (w/o decimal)	byte	1	30, 50, 700 etc.
	short	2	
	int	4	
	long	8	
floating pt type (with decimal)	float	4	3.141, 8.632 etc
	double	8	
Characters →	char	2 → A-Z, a-z, 0-9, #, *, etc.	
True/False →	boolean	Not fixed	

Range of Datatype

1 byte = 8 bits

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

Range of Short

Size = 2bytes = 2x8 = 16bits

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

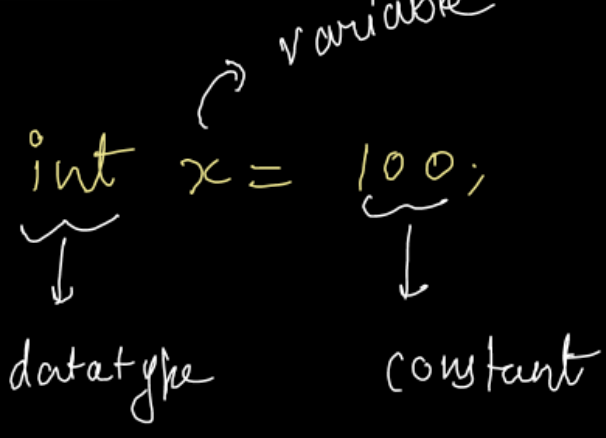
$$\left[-32768 \text{ to } 32767 \right]$$

32768 → 17bit

Data loss

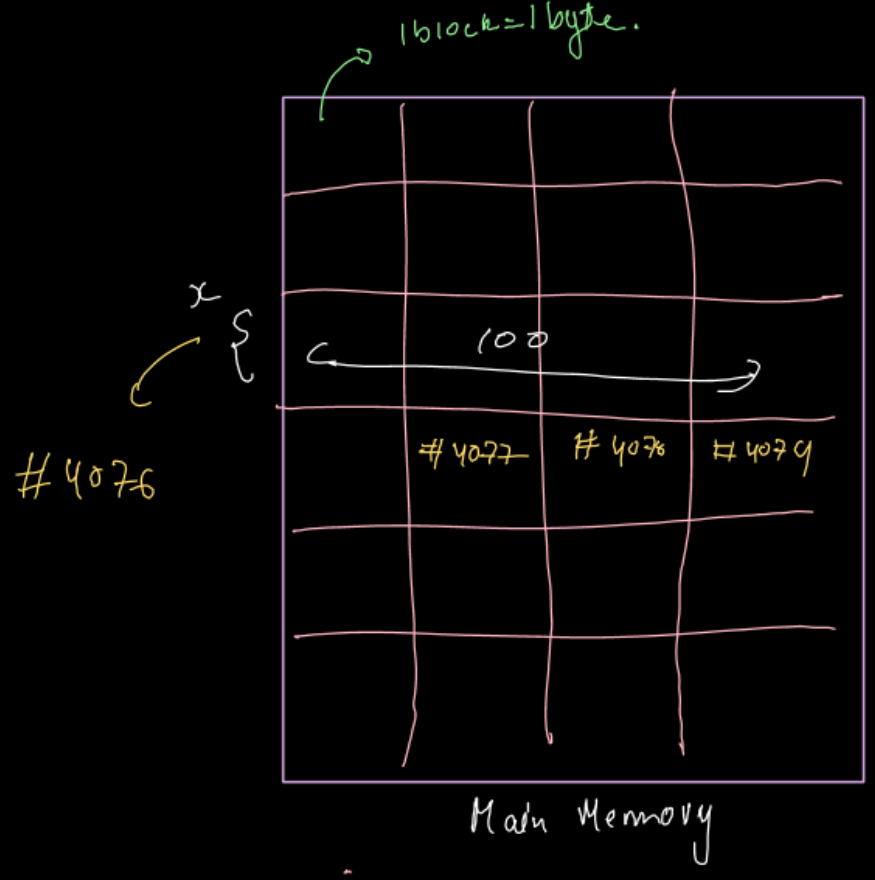
```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x = 100;
        System.out.println(x);
    }
}
```

```
Finished in 62 ms
100
```



int x = 100;

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



```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x = 100;
        System.out.println(x);

        char ch = 'a';
        System.out.println(ch);
    }
}
```

```
Finished in 64 ms
100
a
```

Operators in Java

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x = 100;
        System.out.println(x);

        char ch = 'a';
        System.out.println(ch);
    }
}
```

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x = 100;
        int y = x;
        System.out.println(x);
        System.out.println(y);
    }
}
```

int x = 100;

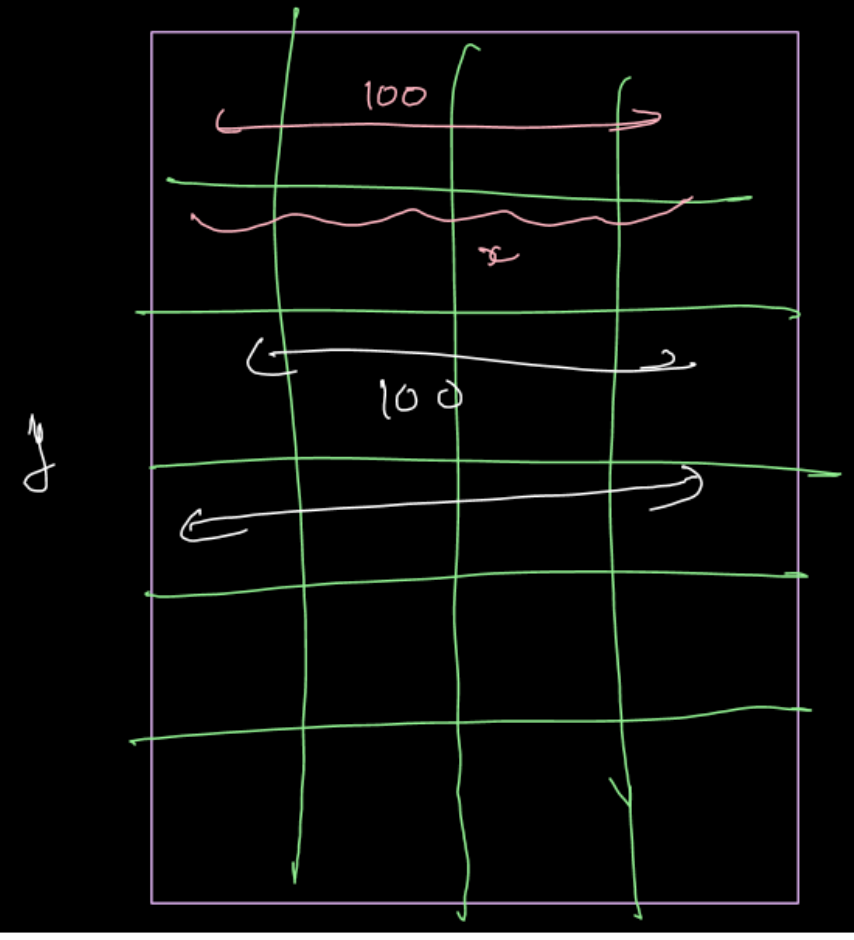
LHS ↓ RHS

assignment operator

RHS gets assigned to LHS

int y = x;

int y = 0;



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

$x = \cancel{0} \; 100$

```
Finished in 50 ms
100
```

* Initialization of values is a must in Java.

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x;
        System.out.println(x);
    }
}
```

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

Arithmetic Operators

\+ plus

\- minus

* multiply

/ divide

% modulus

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 30;
        int res = x / y;
        System.out.println(res);
    }
}
```

10/30 = 1/3 = 0.33...

X

int x = 1.99

= 1

= 0

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 30;
        float res = (float)x / y; //typecasting
        System.out.println(res);
    }
}
```

→ process of converting IDT to another.

integer operator → decimal truncates

x is converted to float 10 → 10.00

float / int = float

10/30 → 0

30) 10 (0

−0

10 → remainder is the ans of modulus.

Quotient is the ans of integer /.

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = 30;
        int res = x % y;
        System.out.println(res);
    }
}
```

```
Finished in 70 ms
10
```

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

```
Finished in 108 ms
256
```

stdin

```
256
```

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int num = scn.nextInt();
        System.out.println(num);

        float f = scn.nextFloat();
        System.out.println(f);
    }
}
```

```
Finished in 164 ms
256
10.987
```

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int num = scn.nextInt();
        System.out.println(num);

    }
}
```

```
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
```

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

```
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

Getting Started (Lecture : 2)

→ Conditional statements
- if - else
- if - else if - else ladder
- switch case

} comparison operators + logical operators

→ functions in Java (with Memory concepts)

→ Loops
- for
- while
- do - while

} continue + break

Questions
- Even or odd 1
- Even or odd 2
- factorial

Conditional Statements

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

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        boolean b = scn.nextBoolean();
        System.out.println(b);
    }
}
```

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

If - Else

```java
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");
    }
}
```

→ true

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

guessNo == 10

LHS — comparison operator — RHS

if LHS = RHS

→ value of this statement is true

else it is false

```
if (condition) {
        ≡              → if condition is true then this executes
} else {
        ≡  } → if condition is false, then this executes
}
```

(condition) → true

```java
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 && guessNo <= 20) {
            System.out.println("You have won the game");
        } else {
            System.out.println("You have lost the game");
        }

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

Greater than or equal to

Less than or equal to

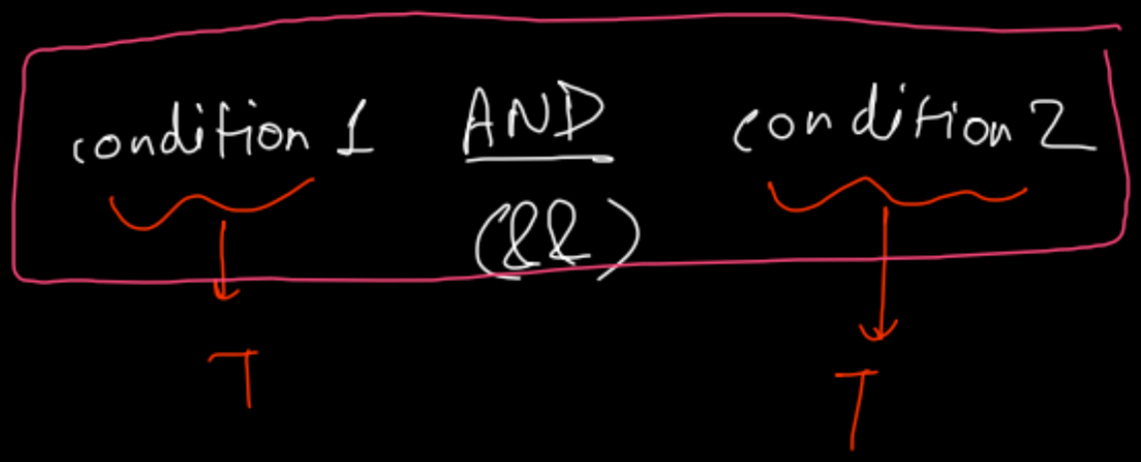
logical AND

HW

> → Greater than

< → Less than

AND



then AND value will be true

else value is false

Logical OR

```
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

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

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

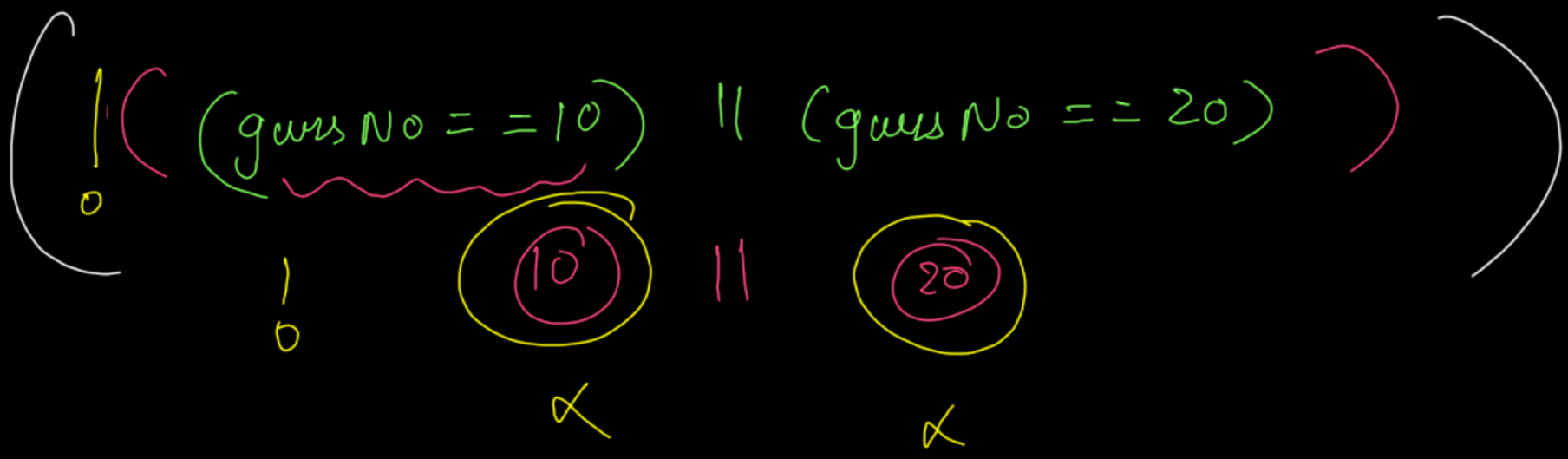
Logical not

! (condition)

↳ it negates the condition

True → false

false → True



Not Equal to

```java
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");
    }
}
```

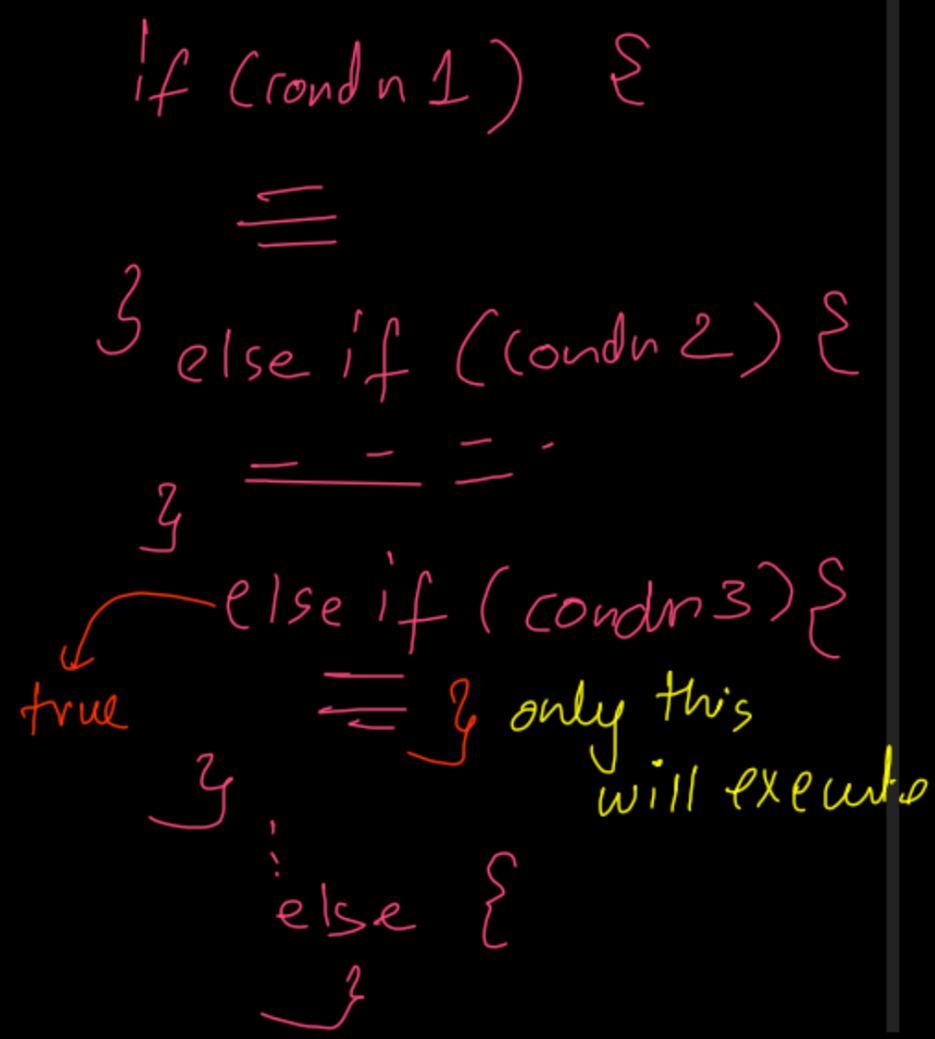
Not equal to

LHS is NOT equal to RHS

Then statement value is true else false.

If – else if – else ladder

```java
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");
    }
}
```



Multible Else if

```java
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.

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

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

        System.out.println("Start");

        if(x == 10) {
            System.out.println("You have won the game");
        } else if(x == 20) {
            System.out.println("You have lost the game");
        }
        if(x == 100) {
            System.out.println("You have won the game");
        }

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

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

for Loop

Print your name 10 times

```java
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
DSA Classes
DSA Classes
DSA Classes
DSA Classes
DSA Classes
DSA Classes
DSA Classes
DSA Classes
```

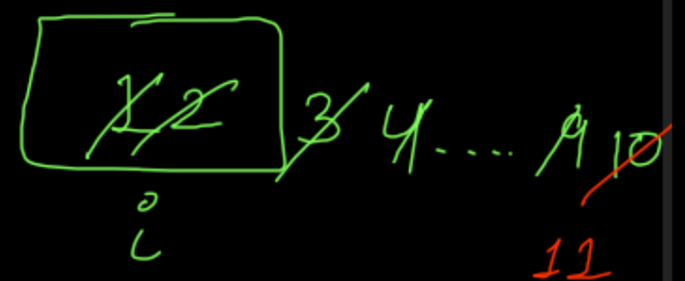
Generally used when exact no of iterations is known.

initial condition

condition

```
for (int i = 1; i <= 10; i++) {
    Syso("DSA");
}
```

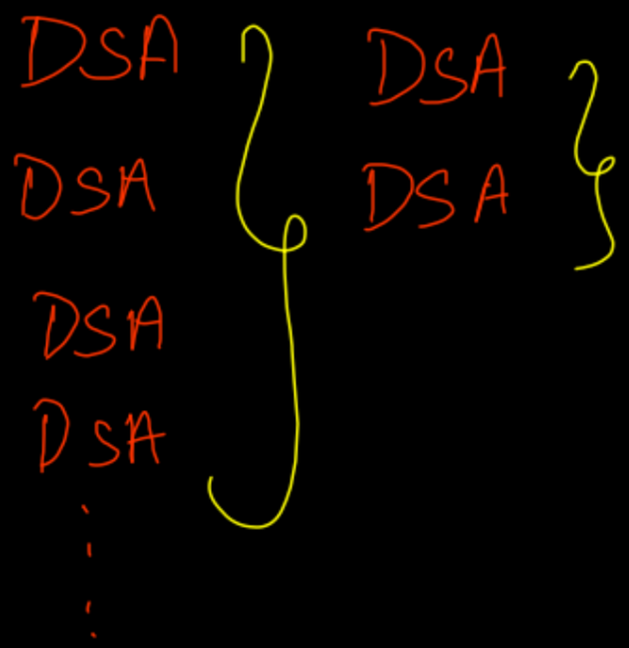
① ② ③



i++ → i = i + 1

(1) + 1

(2)



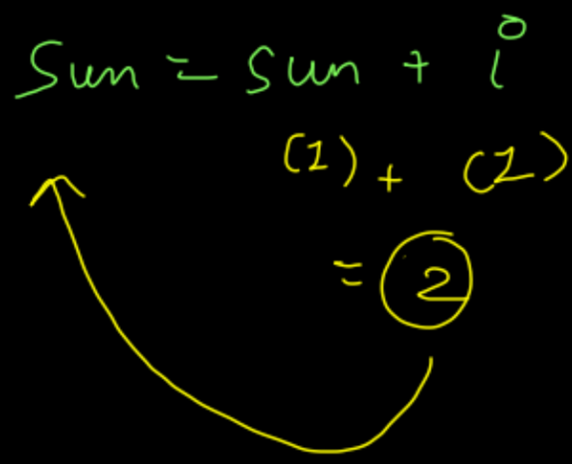
Counting 1 to N

```java
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

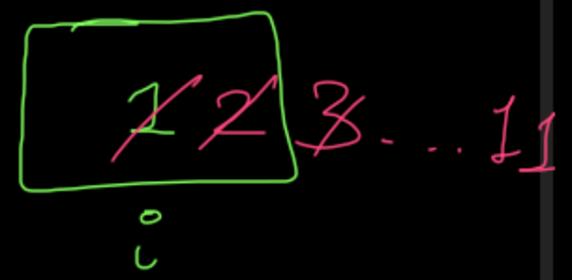
```java
import java.util.*;
public class Main {

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

        int num = scn.nextInt();
        int sum = 0;
        for(int i=1;i<=num;i++) {
            sum = sum + i;
        }

        System.out.println(sum);
    }
}
```



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

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

        int num = scn.nextInt();
        int sum = 0;
        for(int i=1;i<=num;i++) {
            sum += i; //sum = sum + i
        }

        System.out.println(sum);
    }
}
```

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

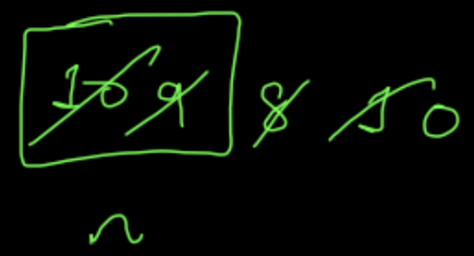
```java
import java.util.*;
public class Main {

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

        int i = 1; → initialization
        while(i <= 10) { → condition
            System.out.println("DSA Classes"); } ②
            i++; ③
        }
    } →
}
```



DSA
DSA
DSA
.
.
.

Count N to 1

```java
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)
        }
    }
}
```

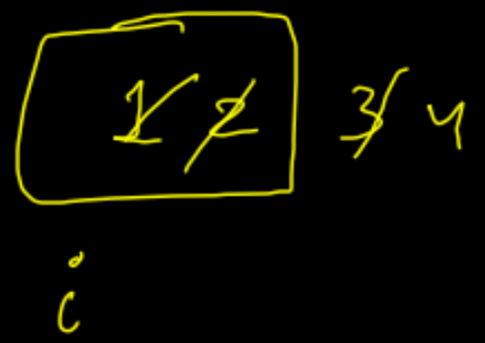


10
9
⋮
1

n-- → n = n - 1

Do while Loop

```java
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
    }
}
```



```java
int i = 11;

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

0

$11 \leq 10 \rightarrow$ false

0x

Loop not executed

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

1

DSA

i = 12

$12 \leq 10$ ✗

Continue in Loops

```java
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++) {
            if(i % 3 == 0) {
                continue;
            }
            System.out.println(i);
        }
    }
}
```

① ③ ② ↓ Skips the current iteration

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

1 2 ~~3~~ 4

1
2
4

i % 3 == 0

Break in loops

```java
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++) {
            if(i % 3 == 0) {
                break;
            }
            System.out.println(i);
        }
    }
}
```

break from the loop
i.e exit the loop.

Switch Case

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

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

        switch(num) {
            case 10:
                System.out.println("Option 10");
                break;
            case 20:
                System.out.println("Option 20");
                break;
            case 30:
                System.out.println("Option 30");
                break;
            default:
                System.out.println("Please select out of 10, 20 or 30");
        }

```

or character or String