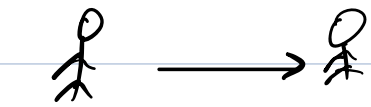
- Introduction
- output
- Operators
- Data types

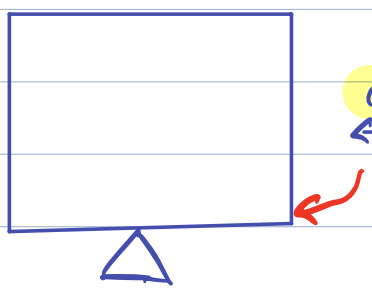
* Computer is ? → Dumb



Sister (elder) → Brother (younger)

get me some water!

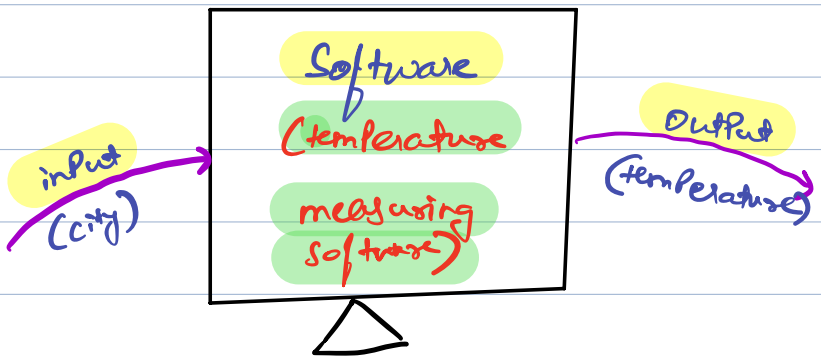
a) get up from bed.
b) go to kitchen
c) get a glass
d) fill glass with water
e) bring the glass to me.



Convert into 0's 1's ← Compiler ← Java / C / C++ / JS / Python etc.

→ Can no we shall ? → rules for english / Grammar

for java as well → you need to follow rules. / Syntax.



leetcode ide → google

writing code

① ↳ IDE : Integrated development environment ? Online editors

↳ Eclipse // Intellij // VSCode

install java.

② output Syntax/rules

↳ System.out.println(10);

↳ next line

↳ System.out.println(7);

Press enter after Printing

↳ System.out.print(7);

Ex:

```
System.out.Print (7+10);
System.out.println (7*10);
System.out.println (10/5);
```

```
17 70
2
```

→ Break till 9:15 PM

↳ I want to print numbers as well as characters!

9 Hello ?!

↳ I want to do mathematical operation with characters!

(10+9)
↳ 19 Hello

* operators

↳ + - * / → BODMAS

5 + 5 ÷ 5 ⇒ 6

↳ Myth busters with SK Sir Part 1.

5 * 5 ÷ 5 ⇒ 5

BODMAS

Rank 1 : Bracket ( )

Rank 2 : Divide / | multiply *

↳ Computer will process left to right.

Rank 3 : add / Subtract

Ex 1: 4 + 3 * 6 - 7 / 2

4 + 18 - 7 / 2 → division in Computer will remove everything after decimal.

4 + 18 - 3

22 - 3 = 19

* Data types

↳ Numbers, Char, double, boolean (true / false) etc.

Integer — ex: 10, 11, 100 etc.

decimal — ex: 5.8, 5.3, 100.2

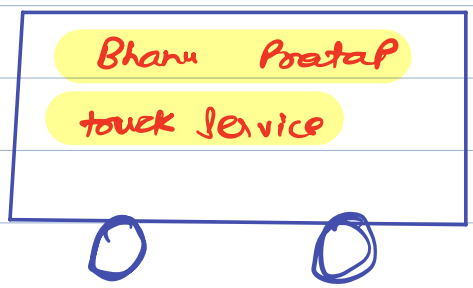
Story:

A B

1 washing machine → Small truck

1 bhk shift → mid size truck

3 bhk shift → large " "



Name of Service: Bhanu Pratap

Type of service: Packers & movers

Quality of Container: Small / mid / large

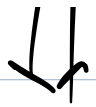
↳ Store → integer

Name of Service → integer

Type of service → Store

Quantity → $10 / 10^9 / 10^{100}$

//



10 point xx

10 Steps

type ↙ name

int temp;

temp = 10;

temp = 100;

Ex: int temp;

temp = 20;

→ temp = 100;

System.out.println(temp); → 100

temp
↳ ~~20~~ 100

Ex: int temp;

temp = 20;

System.out.println(temp); → 20

temp = 100;

→ System.out.println(temp); → 100

temp
↓
~~2~~ 100

Creating integers

1st way

declaration

```
int temp;
temp = 20;
```

initialization

2nd way

```
int temp = 20;
```

both declare & initialize in same line

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

        //Rule1: Terminate your line with semicolon -> comments
        //System.out.println(7);

        //Rule2: Java is a case sensitive lang
        //System.out.println(7);

        //Rule3: Inside the parentheses, we can do maths.
//          System.out.print(7+10);
//          System.out.println(7*10);

//          System.out.println(10/5);

        //Rule4 -> Use double quote to print as it is
        //System.out.println(Hello);
        // System.out.println("Hello");
        // System.out.println("10+17");
        //System.out.println("Hello SK Sir");


        // How does System.out.println(); works?
        // -> In the coming classes we will learn it.

        //Rule5 -> Concatenate different type with + while printing

        // System.out.println(10+9+"Hello");
        // System.out.println(100+9+15+"Hello");
```

```
        // Rule6-> Operators -> BODMAS
        // System.out.println(6+6/6);
        // System.out.println(4+3*6-7/2);

        //Data Type
        // int temp;
        // temp = 20;
        // System.out.println(temp);

        // int temp;
        // temp = 20;
        // System.out.println(tempp);

        //Rule -> You can't create same named variable twice.
//          int temp;
//         temp = 20;

//         int temp;
//         temp = 100;
//         System.out.println(temp);

//         int temp;
//         temp = 20;
//         // int temp;
//         temp = 100;
//         System.out.println(temp);

            // int temp;
            // temp = 20;
            // System.out.println(temp);
            // temp = 100;
```