

---

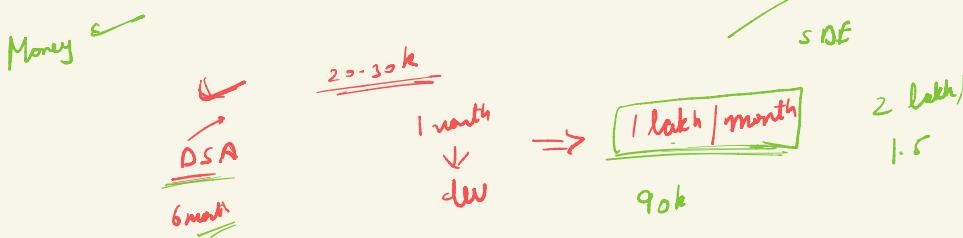
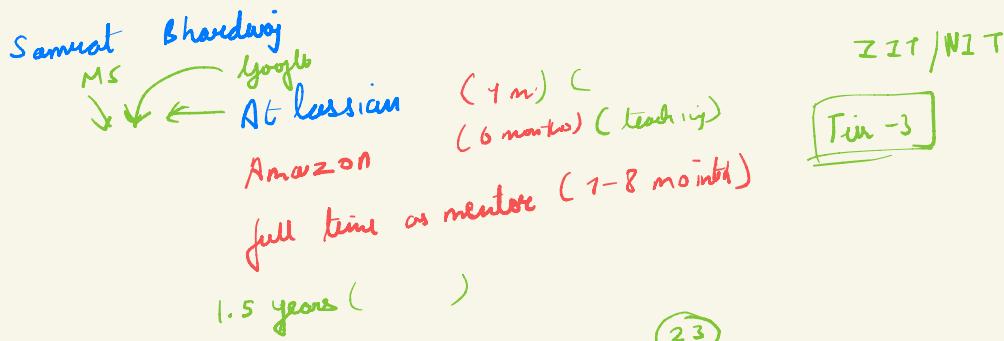
---

---

---

---





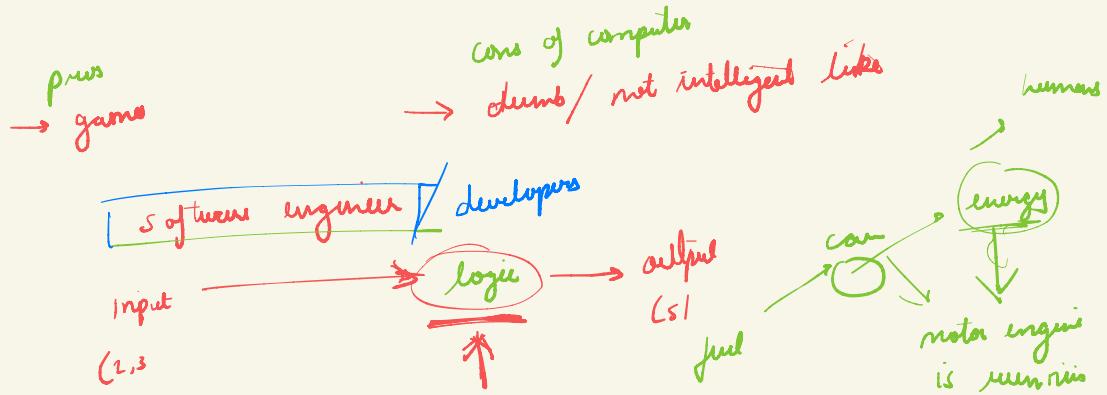
- focus 80% in the class
- review everything after the class (30-40) —
- after 1 month (start giving 3-4 hours of self learning) 1-2 hours

do this for 7-8 months  
(4-5)

40k

Ques what is a computer

what is a machine?

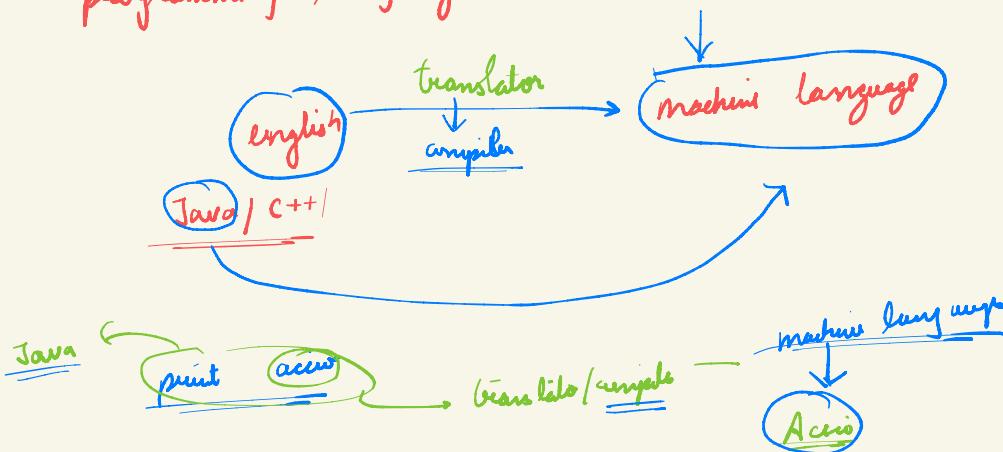


Computer only understands 0's and 1's

↳ machine language

→ 0's and 1's

programming languages



Java → secure, stable, environment-independent

```
public static void main(String[] args) {  
    // how to print in java  
    → System.out.println("Anything here");  
    → System.out.println("Something else");  
}
```

Compiler

Anything here  
Something else

comment

lines which are not executed

↓  
not translated to machine language

RAM

(temporary)

→ random access memory

{ for every program to run, it needs some space

provided by RAM

System.out.println("Hello World");

9:45

computer → machine language

```
public static void main(String args[]) {  
    1) // your code here  
    2) System.out.println("Hello, World.");  
    3) System.out.println("Hello, Java.");  
}
```

Hello, World. Hello, Java

Java → translate → machine language.

```

⇒ System.out.println("Hello, world");
⇒ System.out.println("Acies");
⇒ System.out.print("Job");
⇒ System.out.print("Hello");
⇒ System.out.print("Okay");
⇒ System.out.println("Okay");

```

output area

2	Hello, world
3	Acies
	JobHelloOkay

In ⇒ cursor will move to the next line

Ques 2

```

System.out.println("*");
System.out.println("****");
System.out.println("*****");

```

output console

\*  
\*\*\*\*  
\*\*\*\*\*

```

public static void main(String args[]) {
    System.out.println("*");
    System.out.println("****");
    System.out.println("*****");
}

```

\*  
\* \* \*.  
\* \* \* \* \*  
|

# Variables

↳ that varies.

let  $x=5;$  → integer  
let  $y=10;$

$x=y+1$

let  $x=5.12;$  → decimal

{  $x, y$  }

their values can vary.

$x$  will int value  
int  $x=5;$   
int  $y=10;$   
 $\rightarrow y$  is an integer

$x=y+1;$

# Declaration =  $x \rightarrow \text{int}$

int  $\underline{x};$

int  $\underline{x};$

$x = 5; 12;$

# Initialisation  
 $x = 5;$

int  $\underline{x};$

$x$  will store integer values

declaring  $x$  to be an integer.

```
public static void main(String[] args) {
    // declaration
    1 int x;
    // initialisation
    2 x = 10;
    System.out.println(x);
```

int  $\underline{x};$

declaring  $\underline{x}$  to be an integer.

initialised

$x = 10$

Container

$x$   $\rightarrow$  integer  
name

```
public static void main(String[] args) {
    // declaration
    1 int x;
    // initialisation
    2 x = 10;
    int y = 15;
    System.out.println(x+y);
```

int  $\underline{x};$

$\underline{x} = 10$

$y = 15$

$x + y$

$\checkmark$

$10 + 15$

(25)

System.out.println("Hi " + "Hello");

```
public static void main(String args[]) {  
    1) int var;  
    2) var = 10;  
    3)  
    4) System.out.println("My num is "+var);  
}
```

→ var is declared to be  
an integer.

Var = 10;

int  
Var = 10

## # data-types

primitive data type

most basic

boolean → true  
false

✓ byte → -128 to 127 (2<sup>7</sup>) { 8 bits }  
✓ short → -32,768 to 32,767 { 16 bits }  
✓ int → -2<sup>31</sup> to 2<sup>31</sup>-1 { 32 bits }  
✓ long → -2<sup>63</sup> to 2<sup>63</sup>-1 { 64 bits }

non-permitted data type  
↓  
String array

✓ 5 → 3 bits value  
101  
16  
↓  
10000

float → decimal { 32 bits }  
double → decimal { 64 bits }

int  
double

1 → 00000000000000000000000000000001	1 → 00000000000000000000000000000010	1 → 00000000000000000000000000000011	1 → 000000000000000000000000000000100	1 → 000000000000000000000000000000101
2 → 00000000000000000000000000000010	10 → 000000000000000000000000000000110	11 → 000000000000000000000000000000111	100 → 0000000000000000000000000000001100	101 → 0000000000000000000000000000001101
3 → 00000000000000000000000000000011	110 → 0000000000000000000000000000001110	111 → 0000000000000000000000000000001111	1100 → 00000000000000000000000000000011100	1101 → 00000000000000000000000000000011101
4 → 000000000000000000000000000000100	1110 → 00000000000000000000000000000011110	1111 → 00000000000000000000000000000011111	11100 → 000000000000000000000000000000111100	11101 → 000000000000000000000000000000111101
5 → 000000000000000000000000000000101	11110 → 000000000000000000000000000000111110	11111 → 000000000000000000000000000000111111	111100 → 0000000000000000000000000000001111100	111101 → 0000000000000000000000000000001111101
6 → 000000000000000000000000000000110	111110 → 0000000000000000000000000000001111110	111111 → 0000000000000000000000000000001111111	1111100 → 00000000000000000000000000000011111100	1111101 → 00000000000000000000000000000011111101
7 → 000000000000000000000000000000111	1111110 → 00000000000000000000000000000011111110	1111111 → 00000000000000000000000000000011111111	11111100 → 000000000000000000000000000000111111100	11111101 → 000000000000000000000000000000111111101

10 → 1010