

# # Programming Languages

Set of Instruction

C, Java, C, C++, Python,

shares ideas,  
opinion, both  
communicators align  
one common lang.

Maggi

specific  
task's

- Have a pan / Add water
- Boil water / Add maggi
- Add masala
- stir it for 2 mins
- Serve

# develop desktop Application .

# websites

# Mobile Application

## Game

# # Types of Programming Language

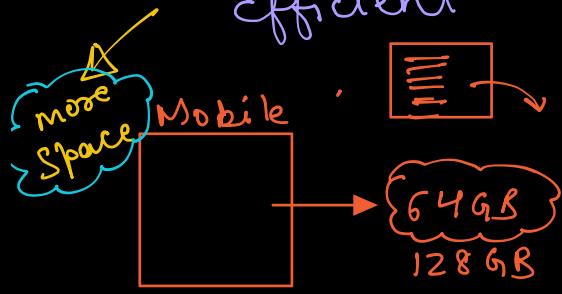
## High Level Lang

→ programmer friendly language.

- Java, C++, Python

→ English

→ Less memory Efficient



When you write a code

- less time to start
- less memory

- Easy to Understand
- Easy to debug
- Simple to maintain

## Low level lang.

→ Machine Friendly



→ execute your task

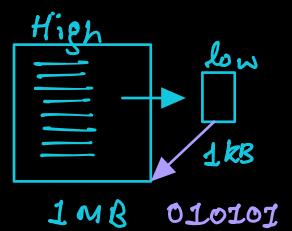


→ More Memory Efficient

→ Low Level

1 kB

less

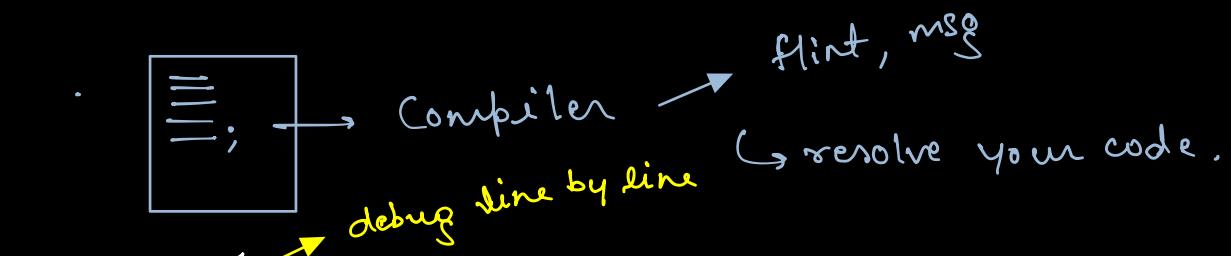


→ More Time Efficient

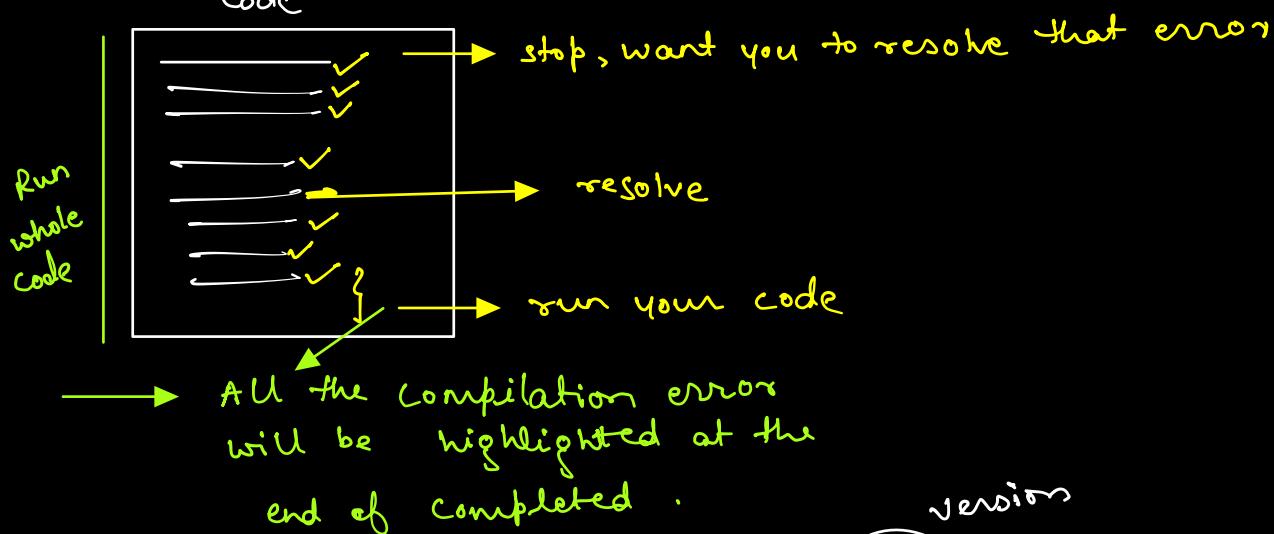
→ More memory Efficient

- Difficult to Understand
- Complex to debug
- Complex to maintain

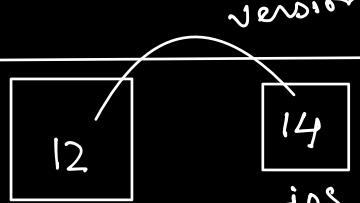
Grammerly → finding , spelling mistakes ,  
Grammer mistakes ,  
tense correction .



### Interpreter Vs Compiler

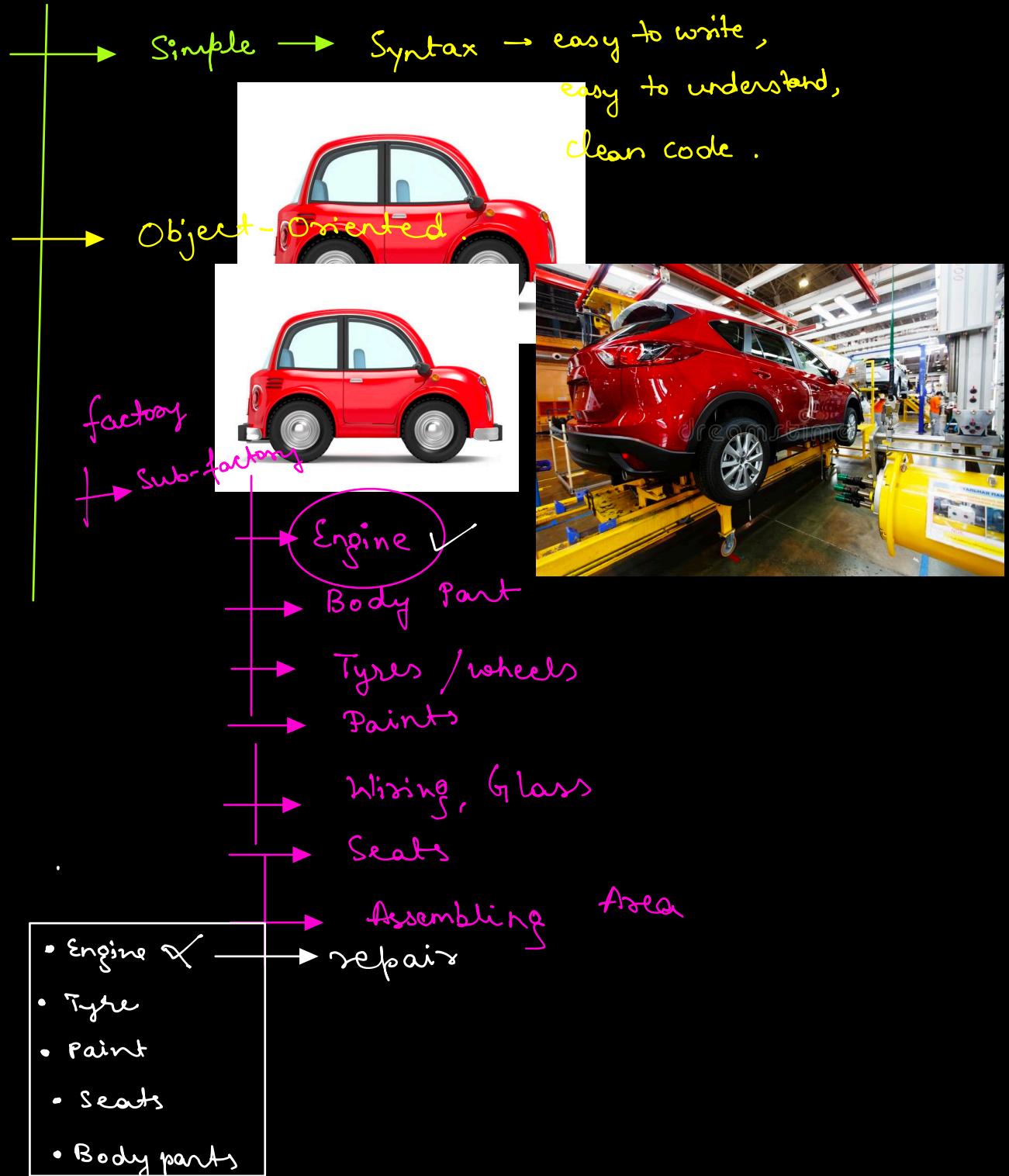


### # What is Java ?

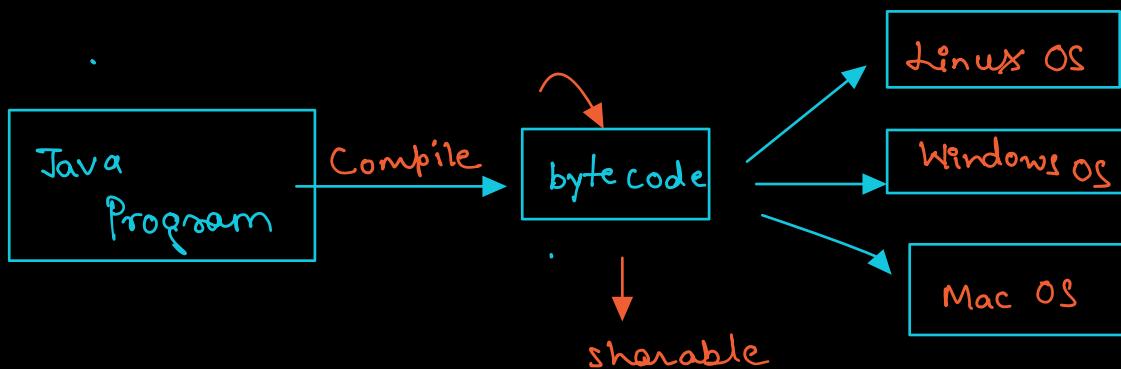


- High level lang. ,
- Object - oriented , secure programming lang. ,
- James Gosling at Sun Microsystems
- In 1991 → OAK → 1995 → Java 7 {Java 8} , Java 13 , Java 15
- 2009 → Oracle → dependencies , library ↗ Stable

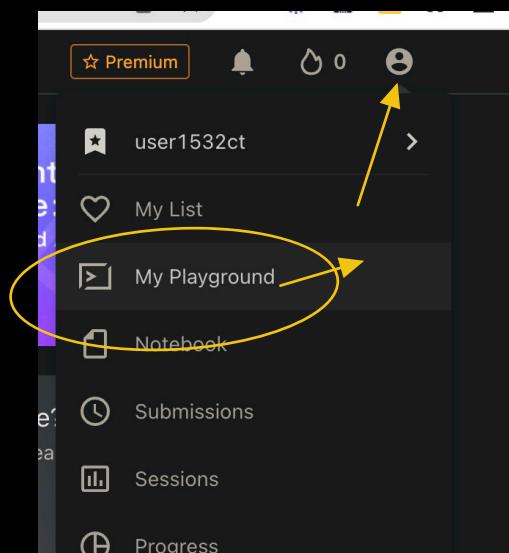
## # Features of Java



# Platform-independent



 Tips → programming → Insite the code more and more.



Boiler plate

A screenshot of a Java code editor showing the following code:

```
public class Main {
    public static void main(String[] args) {
        // → C → code → Run loop
    }
}
```

A green curly brace on the left covers the entire class definition. A green arrow points from the handwritten note "Boiler plate" to the brace.

→ Qualification

*pointing*

```
public static void main(String[] args)
    System.out.println("Hello World");
```

*full stop*

*using*

*output*

*End line*

*debug*

*Compiler*

*Output*

*: compilation error*

Finished in N/A  
Line 4: error: ; expected [in Main.java]  
System.out.println("Hello Geekster")  
^

# *print()* vs *println()*

*print()*      *println()*

- Hello World Hello

→ horizontal

→ Hello

Hello World

```
> class Main {  
public static void main(String[] args) {  
    System.out.println("Hello Geekster");  
    System.out.println("Hello World");  
}
```

→ Hello Geekster  
Hello World

```
> class Main {  
public static void main(String[] args) {  
    System.out.print("Hello Geekster");  
    System.out.println("Hello World");  
    System.out.println("hello");  
}
```

Hello Geekster Hello World  
→ hello  
print() ≡ println()  
'\\n'

```
> class Main {  
public static void main(String[] args) {  
    System.out.print("Hello Geekster\\n");  
    System.out.println("Hello World");  
    System.out.println("hello");  
}
```

## # Hackerrank Challenges

```
import java.util.*;  
public class Solution {  
    public static void main(String[] args) {  
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */  
        System.out.println("Hello World. I am here.");  
    }  
}
```

```

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print out
     System.out.println("Hello");
     System.out.println("World.");
     System.out.println("I");
     System.out.println("am");
     System.out.println("here.");
}

```

```

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print
     System.out.println("*****");
}

```

```

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print
     System.out.println("*****");
     System.out.println("*****");
     System.out.println("*****");
}

```

```

/* Enter your code here. Read input from
System.out.println("*****");
System.out.println("*");
System.out.println("*");
System.out.println("*");
System.out.println("*****");

```

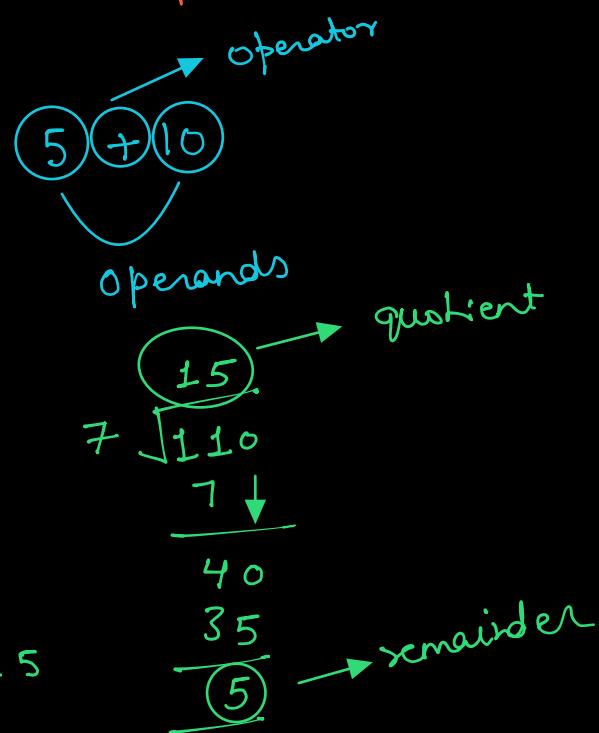
# 8 questions should be completed .

$6 \times 2^3 \rightarrow \text{output};$

## Operators

$\xrightarrow{\quad}$ $\xrightarrow{\quad}$ $\xrightarrow{\quad}$ $\xrightarrow{\quad}$ $\xrightarrow{\quad}$	$+ \quad 5 + 10 = 15$ $- \quad 10 - 2 = 8$ $* \quad 10 * 5 = 50$ <del><math>/ \quad \text{division} \quad 110 / 7 = 15</math></del> $\% \quad \text{modulo} \quad 110 \% 7 = 5$
---	---

quo  
rem



- increment operator  $\rightarrow z = 10 \xrightarrow{+1} 11$   
 $z++ \rightarrow$

- decrement operators

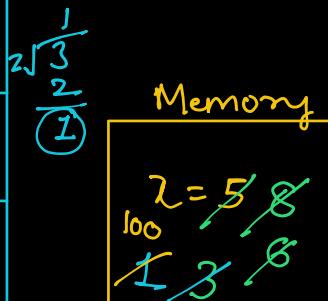
$x = 11 \xrightarrow{-1}$   
 $x- \rightarrow 10$

## # Others Operators

Compact

Expansion

=	$x = 5$	$x = 5 \checkmark$
$+=$	$x + = 3 \xrightarrow{\quad}$	$x = x + 3$
$-- =$	$x - = 2$	$x = x - 2$
$/ =$	$x / = 2$	$x = x / 2$
$\% =$	$x \% = 2 \xrightarrow{\quad}$	$x = x \% 2$
$* =$	$x * = 100$	$x = x * 100$ $1 * 100$



```
int x = 5;
System.out.println(x+=3);
System.out.println(x-=2);
System.out.println(x/=2);
System.out.println(x\%=2);
System.out.println(x*=100);
```