JAVA

★ Boilerplate Code

```
public class Java_File_Name {
    public static void main (~~Datatype~~ Strings args[]) {
        //write our actual code.
    }
}
```

★ Output in Java

```
System.out.print("Hello World");
```

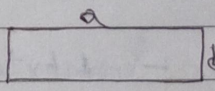
print → Function; "Hello World" → Output; ; → Statement Terminator

- To compile – javac File_name (only 1 time)
- To run – java File_name
- println – For next line (ln).
- \n – Also used for next line.
- Double Quote " " – Strings, we write.
- ~~Single Quote ' ' – Numbers, we write.~~

★ Variables in Java

* Variable is a type of container which contains values that can be changed.

Ex→ a = 10, b = 5



2 * (a + b)

2 → Literals; a, b → Variable

* Literals that cannot be changed, there meanings are universal.

* In Java, we firstly write data types of the variables.

* Syntax- data_type variable_name = "value";
or
value;

Ex⇒ int a = 5;
string name = "Tony stark";

★ DATA TYPES IN JAVA

The different types of data used in java. There are two different

Data types in Java

1. Primitive
(They exist in Java. We not have to create it.)
i.e., byte, short, char, Boolean, int, float, double, ~~etc~~ long.

2. Non-Primitive
(They do not exist. We have to create it.)
i.e., String, Array, Class, Object, Interface

* Java is a typed Language means जब भी Variable declare करते हैं ~~तबतक~~ उससे पहले Data-Type define करते हैं।

* Size of Java Data Types → 1 byte = 8 bits
→ 1 bit = single information

1. byte — 1 byte (-128 to 127)
2. short — 2 bytes
3. char — 2 bytes ('a' to 'z' 'A'-'Z')
4. boolean — 1 byte
5. int — 4 bytes
6. long — 8 bytes
7. float — 4 bytes
8. double — 8 bytes

★ COMMENTS IN JAVA

* Single Line Comment - // Hi
* Multi Line Comment - /* Hello */

★ INPUT IN JAVA

* import java.util.Scanner; (* or)
public class Input { → file name
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
                    → we can change it.
        // write input codes
    }
}

* 1. next
2. nextLine
3. nextInt
4. nextByte
5. nextShort
6. nextFloat
7. nextBoolean
8. nextDouble
9. nextLong

★ TYPE CONVERSION (Widening or Implicit Conversion)

It means converting a data type to another data type.

* Conversion happens when:
  a. type compatible i.e., we can convert numbers to number not numbers to strings.
  b. destination type > Source type
  i.e., Conversion from long to int isn't possible but conversion from int to long is possible.
* byte -> short -> int -> float -> Long -> double
* ~~boolean -> char~~

★ TYPE CASTING (Narrowing or Explicit Conversion)

This is a type of conversion in which we forcefully do conversion, in reply java gives "lossy data" error but then also we perform conversion.

Ex → float a = 25.0; } loss of data
✗ int b = a;
✓ int b = (int) a; // Type Casting.

* If we want to conversion b/t char & int then in int it will return ASCII value of a specific character.

★ TYPE PROMOTION IN EXPRESSIONS

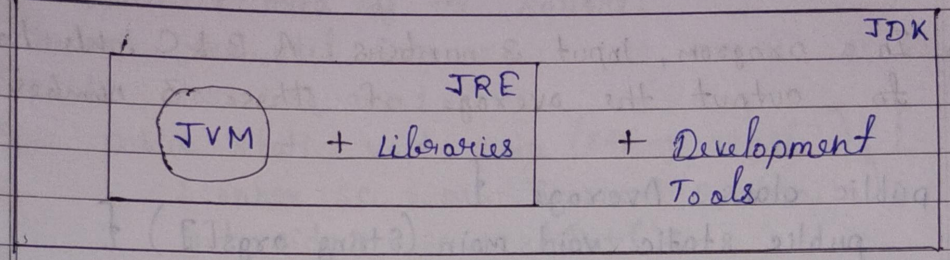
* Java automatically promotes each byte, short or char operand to int when evaluating an expression.
* If one operand is long, float or double the whole expression is promoted to long, float or double respectively.

Ex → char ch = 'a';
short b = 50;
a + b // Firstly, a & b will be converted in int then operation will be done.

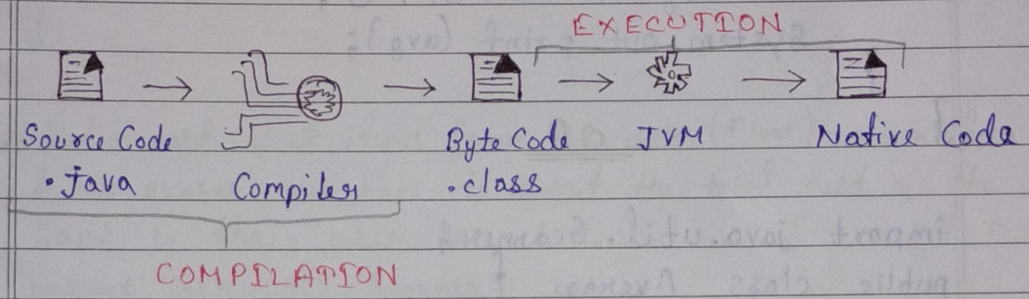
Ex → // WRONG
byte b = 5;
a = b * 2; ⟹ { byte a = b * 2 (becoz Type Promotion)
Error. 99 * 2 = 198.

// RIGHT
byte b = 5;
a = (byte)(b*2); ⟹ { byte a = 5×2 = 10.

★ HOW IS OUR CODE RUNNING?



* JVM – Java Virtual Machine
* JRE – Java Runtime Enviroment
* JDK – Java Development kit.



Java is a portable language i.e., it can be done in all Windows, MAC, Linux. If we write code on windows then it can be run on MAC & LINUX easily.