CORE JAVA

Java: It is Object Oriented Programming language which has developed by "SUN MICROSYSTEMS" & later on it was acquired by Oracle.
• Java is secured & "platform independent" language.

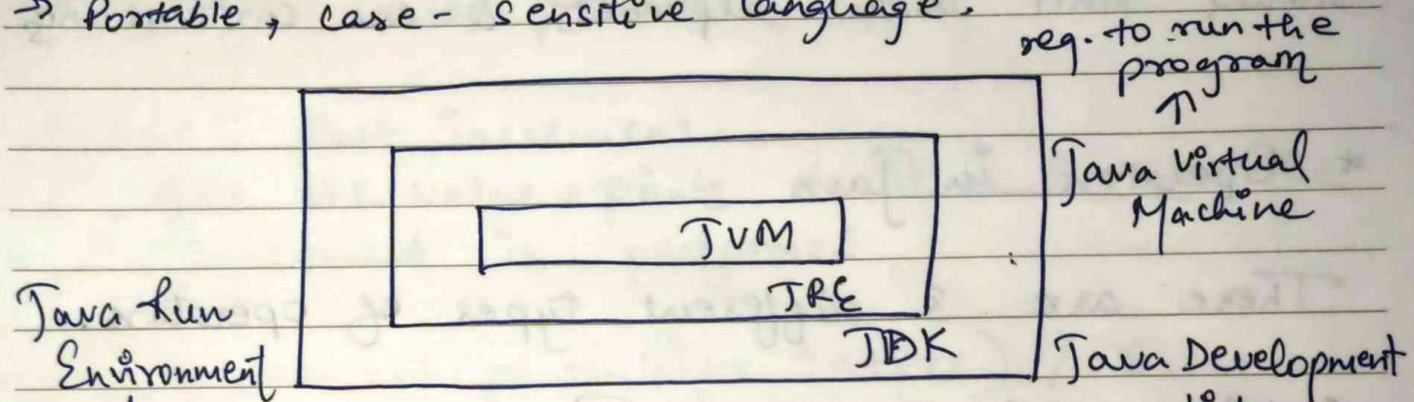
OOP language because it follows all OOPs concept. such as inheritance, polymorphism etc,.

* Applications of Java

1. Web Apps such as Javatpoint.com, IRCTC etc,.
2. Desktop Apps. such as media playes, antivirus
3. Enterprise Apps. such as banking apps.
4. Mobile
5. Smart Card.
6. Robotics, Games etc,.

* Features of Java

→ Platform independent & Open Source
→ Object Oriented lang.
→ Simple, Robust, Secured language.
→ Portable, case-sensitive language.



* Structure of Java Program

```
package packagename;
import Statements (Optional);

public class ClassName {

    public static void main (String [] args)
    {
        Statements 1;
        Statement 2;
            ;
    }
}
```

- package name should start with lowercase letter.
- Class name should start with uppercase letter.
- method / function should start with lowercase

NOTE: method name, package name, Class name should not contain spaces & special characters, Should start with alphabets & can contains nos.

* Operators in Java

There are 3 different types of operators.

1. Arithmetic Operators: To peform mathematical operations like addition, subtraction etc,.

- Addition, string concatenation (+)
  int a = 10;
  int b = 20;
  (a+b)
  30.

- Subtraction, negation (-)
  (b-a)
  Output: 10.

- Multiplication (*)
  (b * a)
  Output: 200

- Division (/)
  (b/a)
  Output: 2

- Equal (=)
  int a = 2;

Unary operators:
- Increment (++)
- Decrement (--)
- Remainder (%)
  int a = 8;
  int b = 2;
  (a/b)
  Output: 0.

eg. int a = 10;
int b = 20;
int c = 5;
int i = 10;

System.out.println(i++ + a); //20.
System.out.println(i) //11

→ i++ : post increment.
first the value prints as it is & then the increment is performed.

System.out.println(i++ + a + i + c); //36.
↓ ↓ ↓ ↓
10 10 11 5

System.out.println(i++ + i++ + i + i++); //45
↓ ↓ ↓ ↓
10 11 12 12

→ ++i : pre increment.
first the value will increment by 1 & then it prints.

eg. Sopln(++i); // 11

Sopln(++i + i); // 22
Sopln(i++ + ++i); // 11 + 13 = 24
Sopln(i); // 13
Sopln(i); // 13.

Sopln (++i + i-- + --i + i-- + ++i); // 48
(++i → 11, i-- → 11, --i → 9, i-- → 9, ++i → 8)

Sopln (i-- + ++i); // 16
(i-- → 8, ++i → 8)

Sopln (++i + --i + c++); // 22
(++i → 9, --i → 8, c++ → 5)

Sopln (-++i + c); // 13
(-++i → 7, c → 6)

Sopln (i++ + c); // 13
(i++ → 7, c → 6)

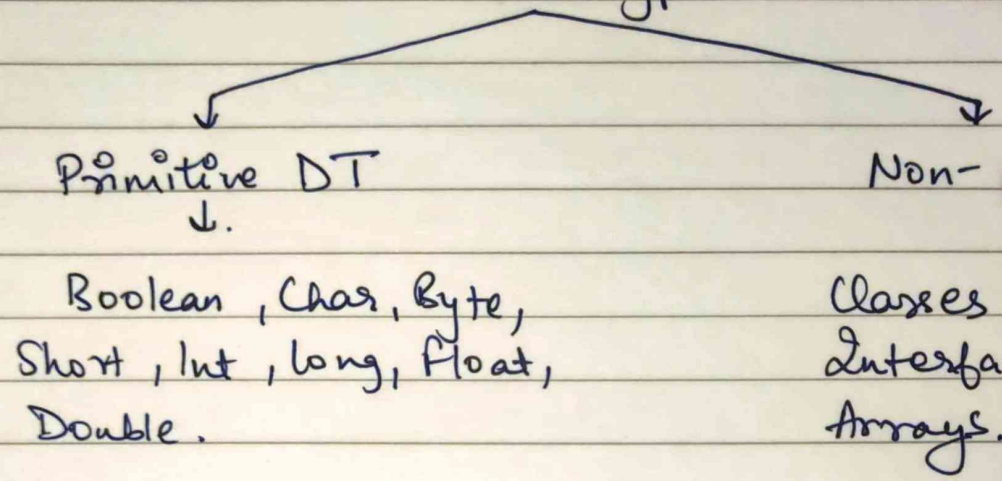
2. Conditional Operators : To check / verify the conditions in a Java program.
Also called Relational operator.
Gives Boolean results (True or false)

- Less than (<)
- Greater than (>)
- less than equal to (<=)
- Greates than equal to (>=)
- Double equal (==) → (comparison) whether both the value are equal or not.
- Not equal (!=)

3. Logical Operators : To perform the logical operation
- AND (Any 1 condition false results false) &&
- OR (Any 1 condition true results true) ||

```
int a = 15, b = 20, c = 5, i = 10;
Sopln((a!=b && a > (c+i)) || (a>c && b>=c+i));
```
(a!=b → True, (c+i) → false, false; a>c → T, b>=c+i → T, True; True // True)

* Data Types : We use datatype to preserve or allocate space for the variables into memory.



Boolean, Char, Byte, Short, Int, long, Float, Double.

Classes, String Interfaces, Object Arrays.