JAVA :-

- It is HLL & OOP language
- C based language

Program - Set of instructions.

Machine Language - Computer's native language (Binary form)

Assembler - Convert Assembly language to Machine language

Note :- Compiler & Interpreter used translate HLL to ML.

JDK - Java Development Kit
- Set of programs that enable us to develop program
- Contain JRE, JVM (It execute java program in other machine)

IDE - Integrated Development Environment

A program that allow us :
- write source code
- Compile
- Debug
- Build
- Run

=> Class - Blueprint to create object
object - An instance of a class

Ex:-
class class_name {
    code block
}

=> Method - Group of instruction to do specific task.

return_type method_name(parameters)
{
    code block.
}

=> Access Modifiers:- Specify how to access classes & methods.

- Public
- Private
- Protected
- Default

=> Package - A container for class

=> Println():-

System.out.println("hello");
// hello

=> Print():

System.out.print("123");
System.out.print("456");

//123456

=> System.out =>

out - It is an object of 'PrintStream' class

System - It is a class

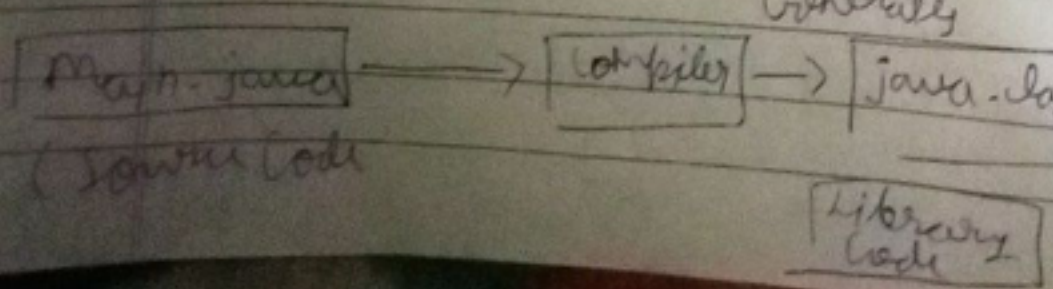
Ex:-

System.out.println(4) // 4
" " " (5+2) // 7

=> To run program in CMD is

First make a sample program with same name as class name + .java

Then open cmd :-
→ javac Main.java
→ java Main



=> Comments - It is used to write note, error

i) Singel line - //
ii) Multi line - /* */

=> Public Access Modifies :-

Access level is everywhere
- Inside a class
- Outside a class
- Inside the package
- Outside the package

=> Private AM :-

Access level is only inside the class.

=> Static Non Access Modifier

You can access field/method using the class name.

=> Command Line Arguments :-

Data given to the main method.

```
S.O.P(args[0]);
// (args[1]);
// (args[2]);
}
```

- Variable - A container that store some data.
  - Each variable has specific types.
- Constants - A variable whose value can not be changed.
- To define a constant we use final keyword.

  final TYPE NAME = VALUE;
- Identifiers - They are the names that identify the elements in a program.

  - Name of classes
  - " " methods
  - " " variables

Rules:

- Can contain letter, digits, underscore & dollar sign.
- Must start with letter, underscore or an dollar sign, Not digits & not contain spaces.

=> Type Conversion :-

byte → short → int → long

=> Size of different integer data types :-

byte - 1B → 8b
short - 2B → 16b
int - 4B → 32b
long - 8B → 64b

=> Calculating Range :-

$[-2^{no\ of\ Bits-1}, (2^{no\ of\ B-1}) - 1]$

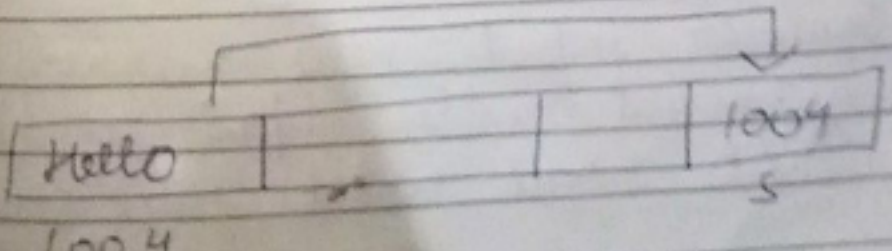
Ex:- short 16b

$[-2^{15}, 2^{15} - 1]$

$: [-32768, 32767]$

=> Reference Type - The variable contains the address of value variable references the value.

String s = "hello";



=> Instantiating an object :

ClassName ObjectName = new
ClassName(Parameters);

Ex:-
School s = new School();

=> Immutable objects :

- Objects whose contents cannot be changed
- Immutable object is a object whose content can not be changed.
- Immutable objects are created from immutable class.
- String class in Java is immutable.

=> Scanner class :

It is used to take input from user.

// Scanner input = new Scanner(System.in);
// System.out.println(input.next());
It read until find space.

System.out.println(input.nextLine());
It read whole line.

Q Write a program for take input from user favorite number?

```
import java.util.Scanner;

public class Main {

    psvm(String[] args)
    {
    Scanner input = new Scanner(System.in);
    System.out.println("Enter the no");
    int a = input.nextInt();
    System.out.println(a + "Favourite no");
    }
    }
```

=> Literals - instant values that appear directly in a program.

Ex -

```
s.o.p('b');
int i = 4;
```

=> Assignment Operator - An operator to assign a value/exp to a variable

Ex -

```
int i1 = 1;          // 1
double d = 5.0;      // 5.0
int i2 = 10;         // 10
i2 = i2 + 10;        // 20
```

=> Arithmetic Operators :-

i) Addition :-
SOP("The sum is:" + (1+3));

ii) Subtraction
SOP("Diff is" + (3-1));

iii) Multiplication :-
SOP("Mul is" + (3*1));

iv) Division
SOP("Div is" + (6/2));

v) Modulo
(x - 5%2 = 1
(Check remainder)

=> Increment Operator :-

Used to increase the variable by 1

i) Pre increment ii) Post increment
++i i++

=> Decrement

i) Pre ii) Post
--i i--

=>

i)

ii)

=>

=> Casting :- Converting datatype to another datatype

i) Implicit Casting :- It happen automatically when converting from a narrower range data type to a wider range datatype

Ex:- double d1 = 4; // int -> double
Convert int to double / float / long

ii) Explicit Casting - It should done by programmer when converting from wider to narrower datatype

Ex:-
Convert double / float / long ⟶ int

Ex:- (datatype) (exp

int d1 = (int) 4.5; // 4
double d1 = 4.5 + (double) 3 // 7.5

=> IF Statement - Used to execute a piece of code based on a condition

if (boolean exp)
statement

=> Switch statement is

Used to execute different cases based on equality

```
switch(exp)
{
case exp-1: statement
    break;
case exp2: statement
    break;
default: statement
}
```

=> Loop - It is used to execute block of code more than once

i) while loop is

```
while ( boolean exp )
{
    Statement;
    Inc/dec;
}
```

ii) do while loop :-

```
do {
    statement;
    inc / dec ;
} while (exp)
```

iii) For loop

```
for (variable ; cond ; change)
  {
    statement
  }
```

⇒ Break - Exist loop regardless of condition.

Continue - skips the rest of loop body