**JAVA:**

* Giving instructions and commands to run the instruction is nothing but programming
* Processor is a combination of Logic Gates
* Computers can understand only [ 0’s and 1’s] are called as Binaries
* We can’t write everything in binary format so we are writing programs
* There are so many programming languages like **C, C++, JAVA, PYTHON, PHP**
* Every programming language has its own implementations

**Ex: If, else, if else**

* Every programming language has its own **COMPILER,** compiler helps to convert the program in to binary format
* If we want to use java on our system, we need to install jdk.
* We have some components in java those are **JRE, JVM, JDK**

**JVM**

**JAVA VIRTUAL MACHINE**

**LIBRARIES**

**JRE**

**DEV TOOLS**

**JDK**

* Here the **java run time environment contains** a component called **JVM,** The JVM helps to converting the program in to computer readable format that is nothing but binary format.
* For example, we have written a program in **c** language. We can’t able to run it in different distributions of Operating system.

1). In windows itself we have different processors **32-bit, 64-bit**

So, the C language is not portable.

* In the case of java, the source code will first convert as BYTE CODE then it will be converted to BINARY, Because of this process we can run the java code in anywhere like macOS, Linux
* The conversion of src code to Byte code happened by using **Dev tools (Java C, Java)**

**VARIABLES IN JAVA:**

* We can store values in variables, variables are said to be as containers, and the values can be changed.
* **SYNTAX FOR ASSIGING A VALUE TO A VARIABLE**

Datatype Variable Name = value;

**Ex:**

Int a = 10;

* When writing programs following naming conventions will be good, for example using

**camelCase (**studentName, goodLuck, studentRollNumber)

* As per the rules it is allowed to write a variable by starting with **\_ $**
* But usually java developers won’t use **\_ $** when writing a variable
* **Java** name can’t be a keyword
* Java itself knows the keywords, some of the keywords are listed below

**Public class**

**Public class void main**

* So, keywords are should not be used as variables

**DATA TYPES:**

**PRIMITIVE DATA TYPES:**

* int -------- > used to store numbers (1,2,3,4,5,6,7,8,9)

int is a 32-bit data type if it exceeds, we can use **long** datatype

if it shorter than 32-bit we can use **short** datatype

* float -------- > used to store decimal points (1.1, 2.3, 4.5, 6.7, 3.14) up to 6 decimal points
* double -------- > its also for decimal values but more than 6 decimal points we can store
* boolean -------- > its for **TRUE or FALSE** conditions (or) **1 or 0** (or) **YES or NO**
* char -------- > to store characters for example (‘malli’, ‘shiva’, ‘krishna’, ‘durga’)

here each individual alphabet is a character, even every symbol in the

keyboard is a character

**TYPE CASTING:**

It is nothing but conversion of datatype into another datatype, values will be changed conventionally so it called as type convention

**Example 1:**

Int a = 8;

float b;

b = a;

**output :**  a = 8 , b = 8.0 (it is direct method and wont effect any error)

**Example 2 :**

float x = 3.14;

int y;

y = x;

**output :** it will throw error and will not work, so instead of this we can write in another way

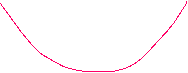
float x = 3.14;

int y;

y = (int) x;

**TYPE CONVERSION ORDER :**

byte , short, int, long, float, double



* the above patterns shows the type conversions, and its only in direct method refer **EX 1**
* we can also use indirect method for any conversion refer **EX 2**

**BASIC INPUT & OUTPUT:**

in java we have system.out.println to print the output

* here ln separates the output for next line
* **EX 1:**

System.out.println(a);

System.out.println(b);

Output : a

B

* **EX 2:**

System.out.print(a);

System.out.println(b);

Output : ab