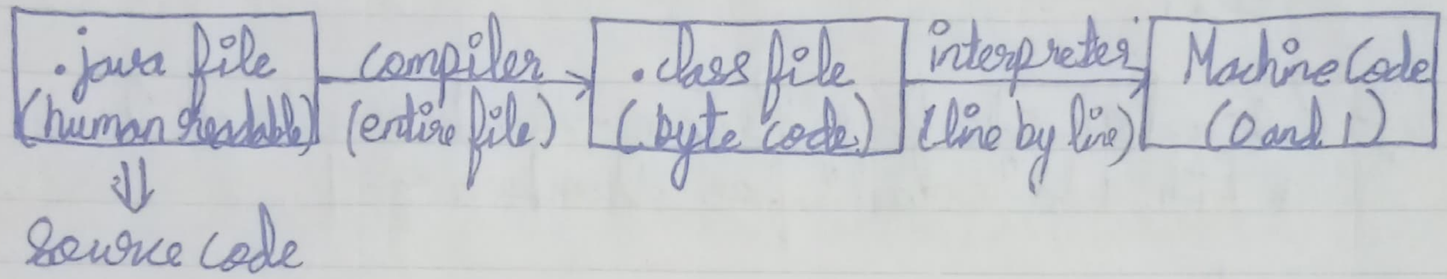
★ Lecture 3 - Intro^n to Java - Architecture and Installation

#1:) How Java Code Executes :-



Q⇒ What is Byte Code?

Ans⇒ It is an ~~ind~~ intermediate language that java uses

Q⇒ Why we need this Byte Code?

Ans:⇒ 1:) This is the reason why java is platform independent. (in C, C++ etc source code convert directly to machine code)

2:) We need JVM (Java virtual machine) to run this byte code in our system.

Note :- 1:) JVM is platform dependent but byte code/Java can run on any system in which JVM is installed. hence platform independent.

2.) Byte code is kind of encrypted form of source code so it is sechure to share Byte code over systems. In that way source code is secure and also can perform tasks on any system.

3.) JVM runs Byte code but not show original source code out of it.

#2.) Some Points on Platform Independence:-

a.) Byte code can run on all OS.
b.) Need to convert source to machine code.
c.) Compiler do above by turning source into executable code.
d.) executable is set of instn for computer.
e.) After compiling C/C++ code we get .exe file which is platform dependent.
f.) In java we get byte code and JVM convert this to machine code.
g.) Java is platform independent but JVM is platform dependent.

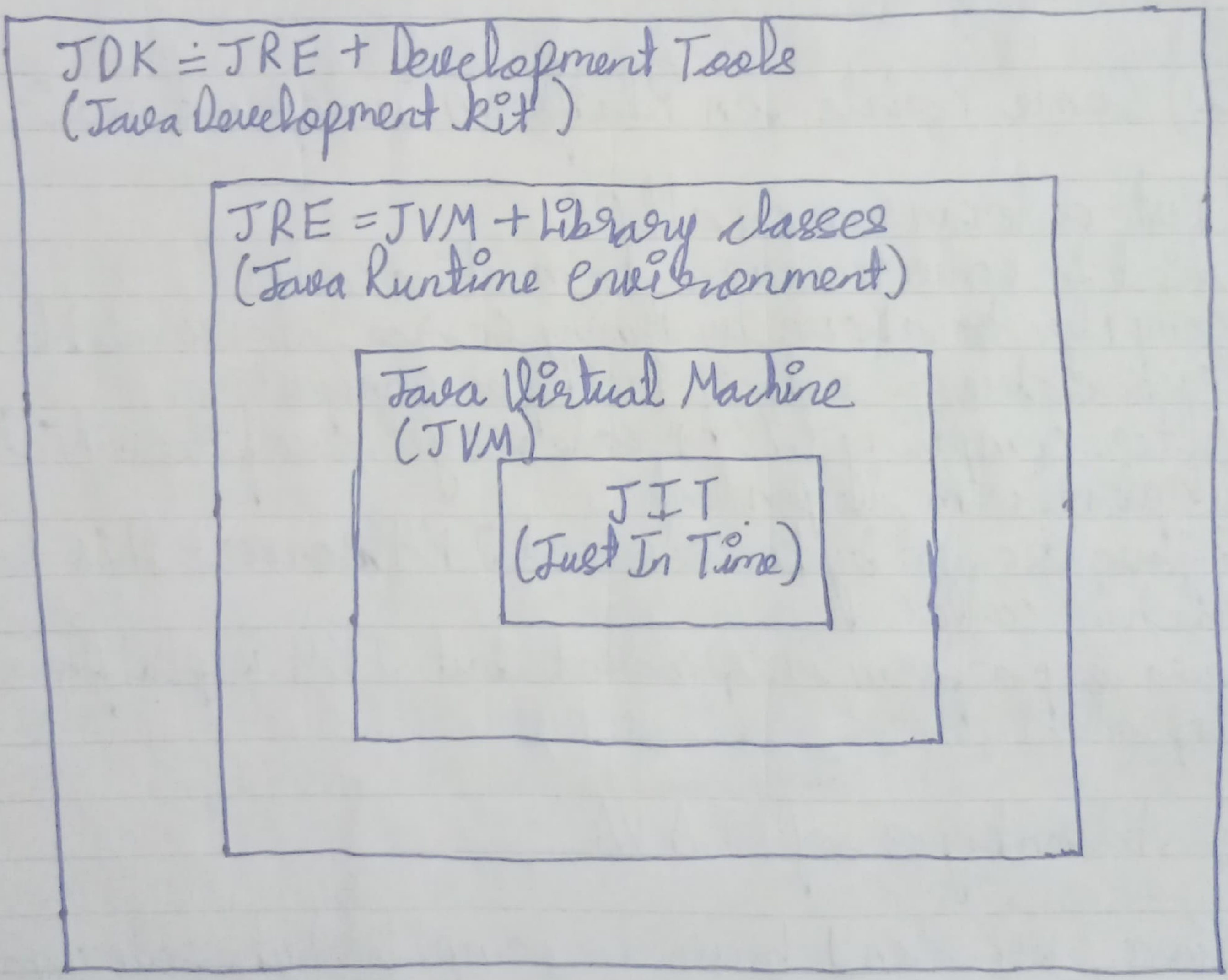
#3.) Advantages of JAVA:-

1.) Since, we don't have to compile byte code again & again for every system on which we want to run it unlike C/C++ where, exe file need to compile on every system, Java is fast during reusability.

2:) More secure due to byte code cos it don't show even source code but can run it.

#4.) Architecture of Java:-

→ JDK vs JRE vs JVM vs JIT



JDK = JRE + Development Tools
(Java Development Kit)

JRE = JVM + Library classes
(Java Runtime environment)

Java Virtual Machine
(JVM)

JIT
(Just In Time)

1:) JDK
(Java Development kit) :-

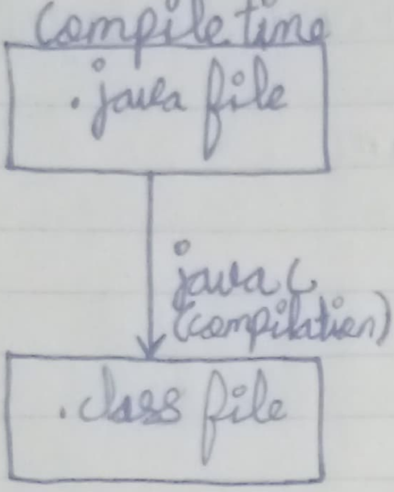
- Provides environment to develop and run the Java program.
- It is a package that includes:-
  a.) develop^n tools - to provide an env^n to develop your program.
  b.) JRE - to execute your program
  c.) a compiler - javac
  d.) archiver - jar
  e.) docs generator - java doc
  f.) interpreter / loader

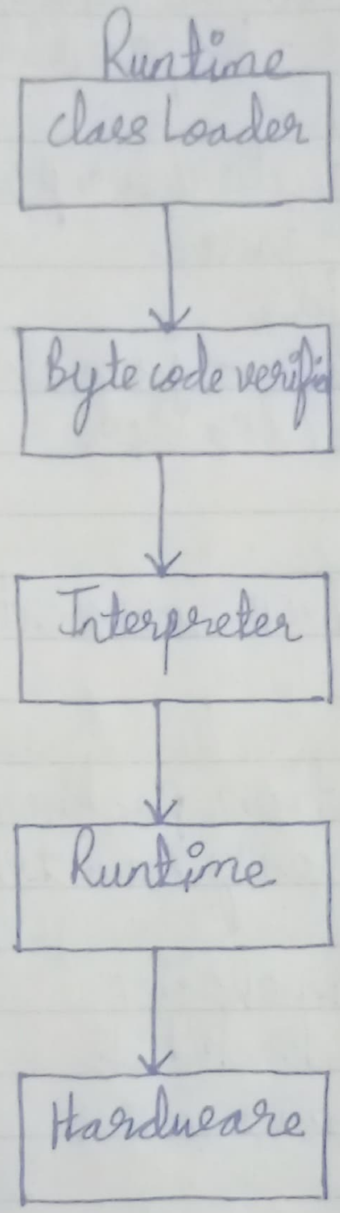
2.) Java Runtime Environment :-
(JRE)

- It is an installation package that provides environment to only run the program
- It consists of:-
  a.) Deployment technologies
  b.) User interface toolkits
  c.) Integration libraries
  d.) Base libraries
  e.) JVM

- After we get the .class files, the next thing happens at runtime:
  a.) Class loader loaded all classes needed to execute the program.
  b.) JVM sends code to Byte Code verifier to check the format of code.

#5.) Explanation of Architecture Components Working :-





JVM Execution

Interpreter:
- Line by Line execution
- when one method is called many times, it will interpret again & again.

JIT:
- those methods that are repeated, JIT provides direct machine code so re-interpretation is not required
- makes execution faster

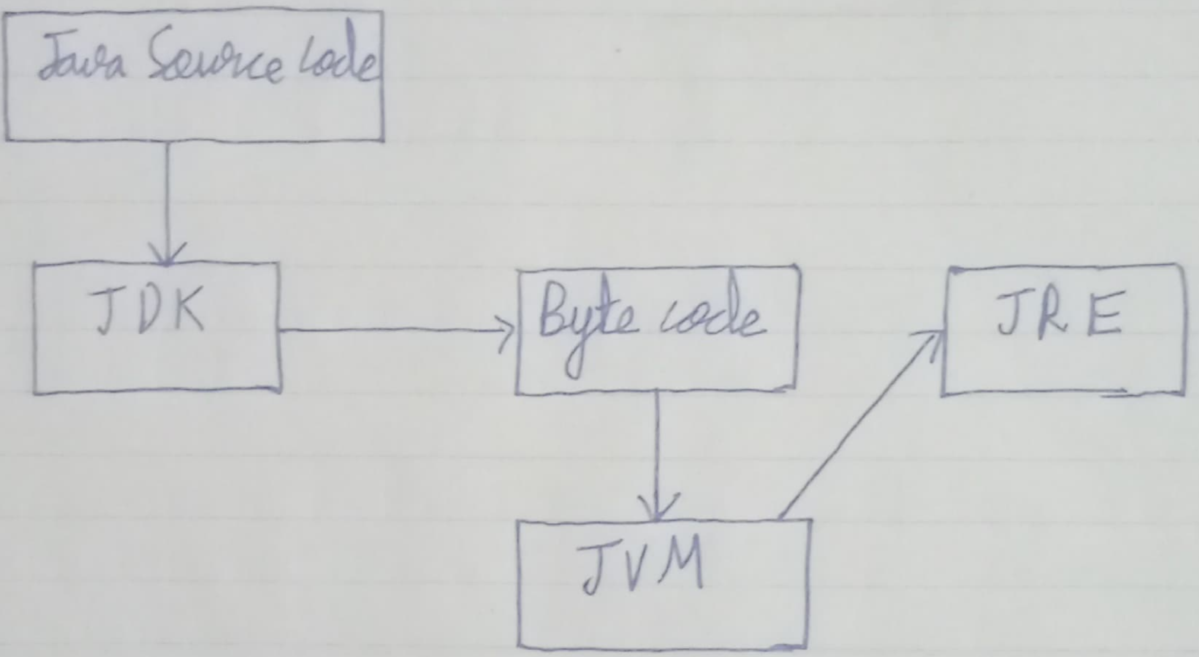
Garbage collector:
- discussed previously

• (How JVM works) class loader
- Loading :-
  a.) Reads .class file and generate binary data
  b.) An object of this class is created in heap
- Linking :-
  a.) JVM verifies the .class file
  b.) Allocates memory for class variables & default values
  c.) Replace symbolic references from the type with direct references
- Initialization :-
  All static variables are assigned with their values defined in the code & static block.

JVM contains the stack and Heap Memory allocations.

#6:) JRE vs JVM :-

Note :- All files (Library files) that JVM required for la particular byte code to convert it into machine code & generating/processing output/input is supplied by JRE


Java Source Code
JDK
Byte code
JRE
JVM