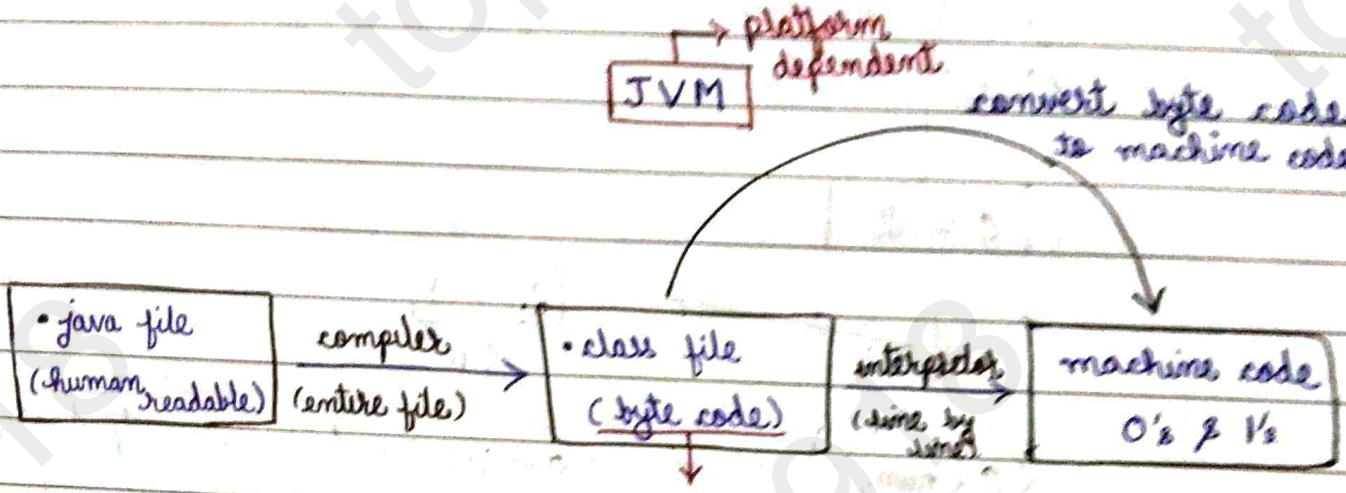
13/09/21

Introduction to JAVA

* How JAVA code executes and more information about platform independence



* Therefore, JAVA is platform independent *

⇒ we can provide this byte code to any system means we can compile the JAVA code on any system.

⇒ But JVM is platform dependent means for every O.S, the ~~exe~~ executable file we get, it has step by step set of instruction dependent on platform.

JDK vs JRE vs JVM vs JIT

JDK (JAVA Development Kit)

↳ provide run time to run and develop JAVA program

JRE (JAVA Runtime Environment)

↳ provides environment to run the program only

- → deployment technologies
- → user interference toolkit
- → integration libraries
- → base libraries

etc.

JDK (JAVA Development Kit)

- → development tools
- → javac → JAVA compiler
- → archiever → JAR
- → docs generator ↳ javac
- → interpreter / loader

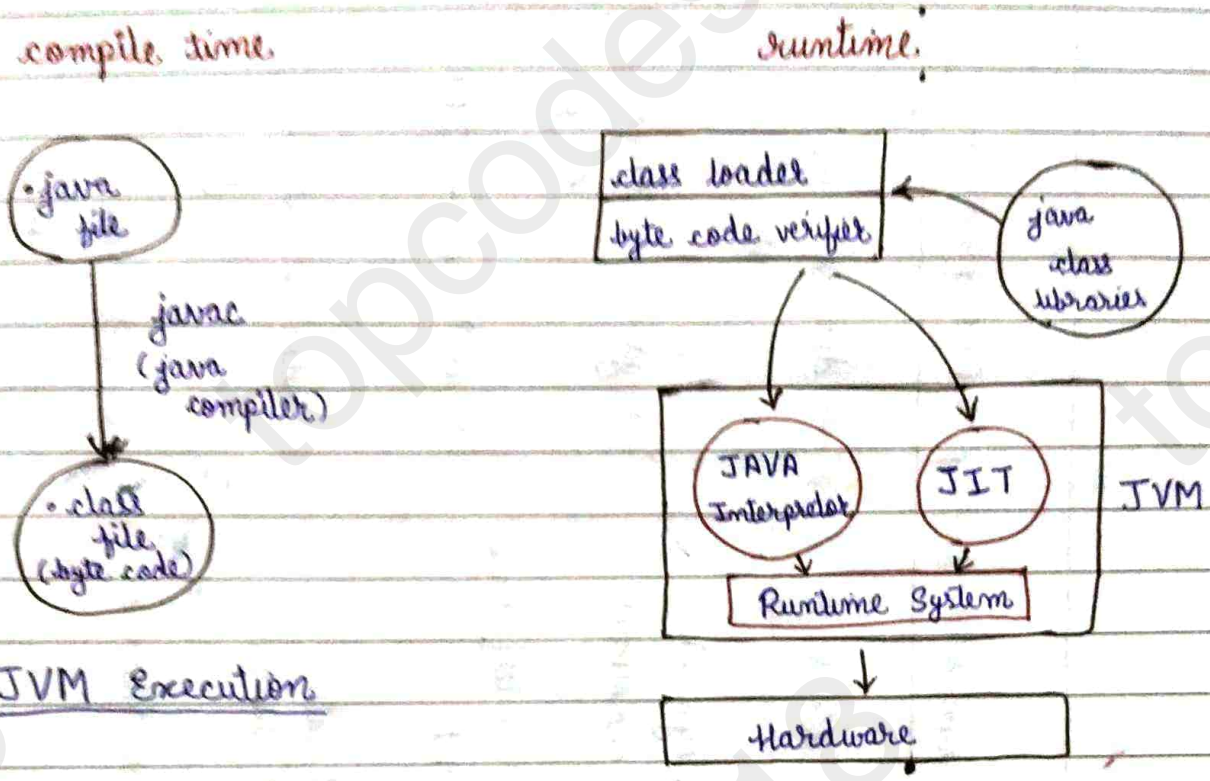
etc.

JIT (Just in Time)

- → Java Interpreter
- → garbage collector

etc

JAVA Development and Runtime Environment



⇒ JVM Execution

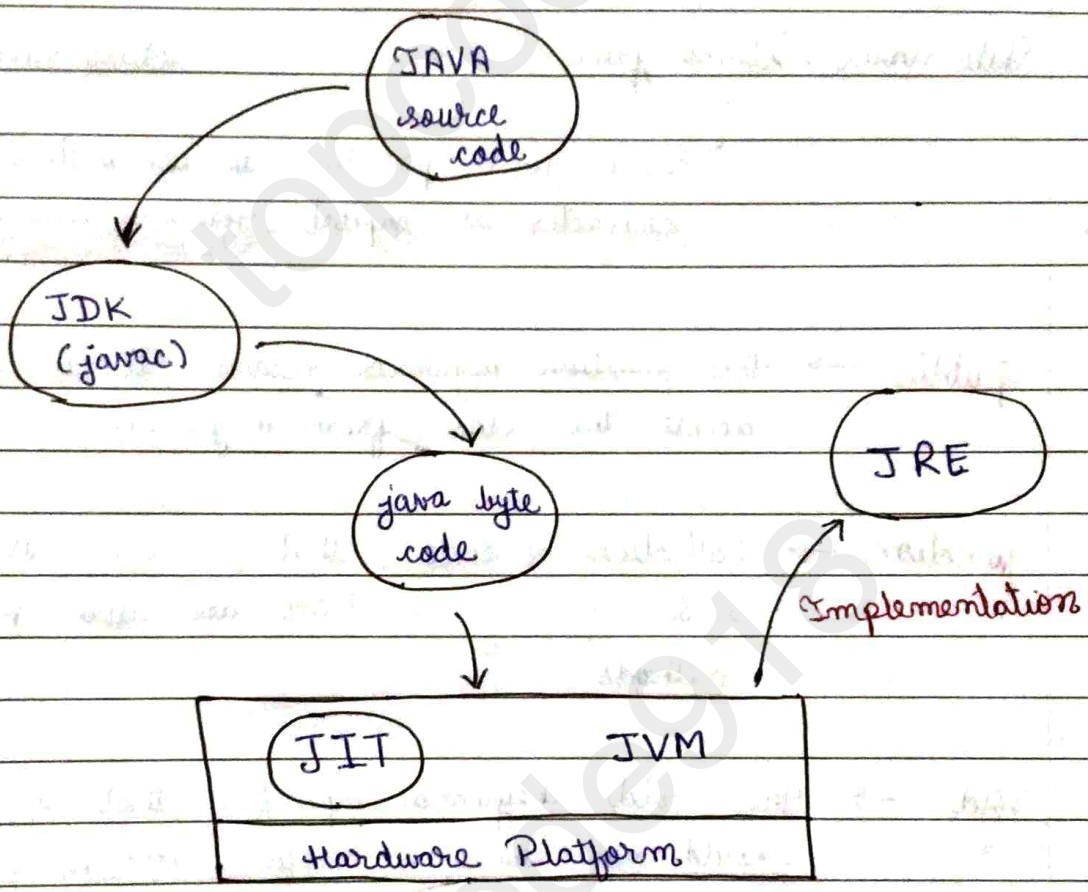
- JAVA Interpretor
  - → line by line execution
  - → when any one method is called many times it will interpret again & again
- JIT
  - → method that are repeated, JIT provides direct machine code so re-interpration is not required.
  - → makes execution faster
- Garbage Collector

* Class loader :-

- loading
  - → read byte code file & generate binary data
  - → an object of this class is stored in heap.
- Linking
  - → JVM verifies .class file
  - → allocate memory for class variable for and default value
  - → replace symbolic reference from the type with direct ref.
- Initialization
  - → all static variables are assigned with their value define in code

__/__/____

Summary


JAVA source code
JDK (javac)
java byte code
JRE
Implementation
JIT
JVM
Hardware Platform