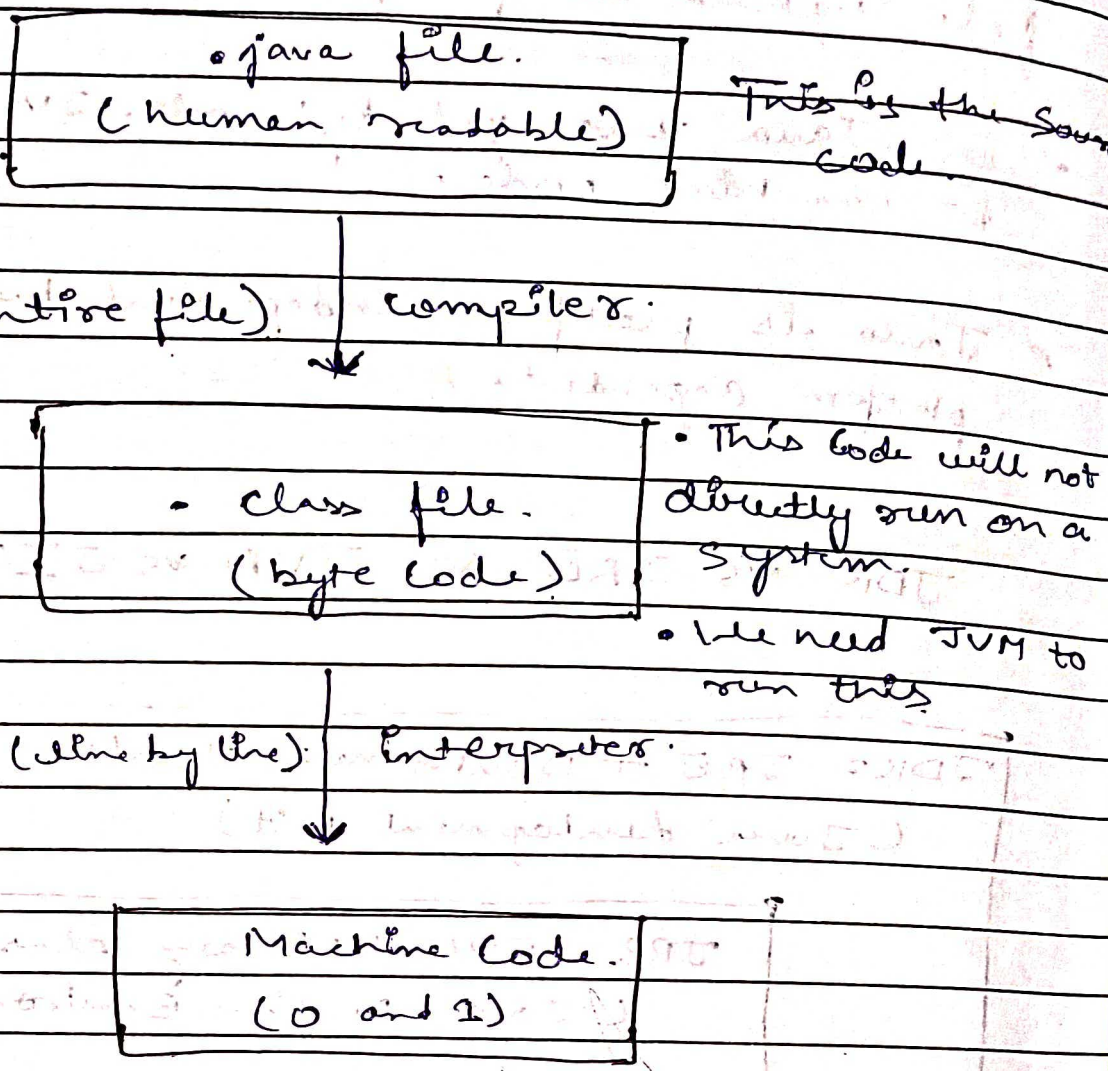
How Java Code Executes.



Java is Platform Independence.

- It means that byte code can run on all operating systems.
- We need to convert source code to machine code. So computer can understands
- Compiler helps in doing this by turning it into executable code.
- This executable code is a set of instructions for the computer.

DATE: / /

• After compiling C/C++ code we get .exe file which is platform dependent.

• In Java we get bytecode, JVM converts this to machine code.

* Java is platform-Independent but JVM is platform dependent.

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)

JDK.

* Provides environment to develop and run the Java program.

* It is a package that includes :

1. Development tools – to provide an environment to develop your program.
2. JRE – to execute your program.
3. a compiler – javac.
4. archiver – Jar.
5. docs generator – Javadoc.
6. Interpreter / Loader.

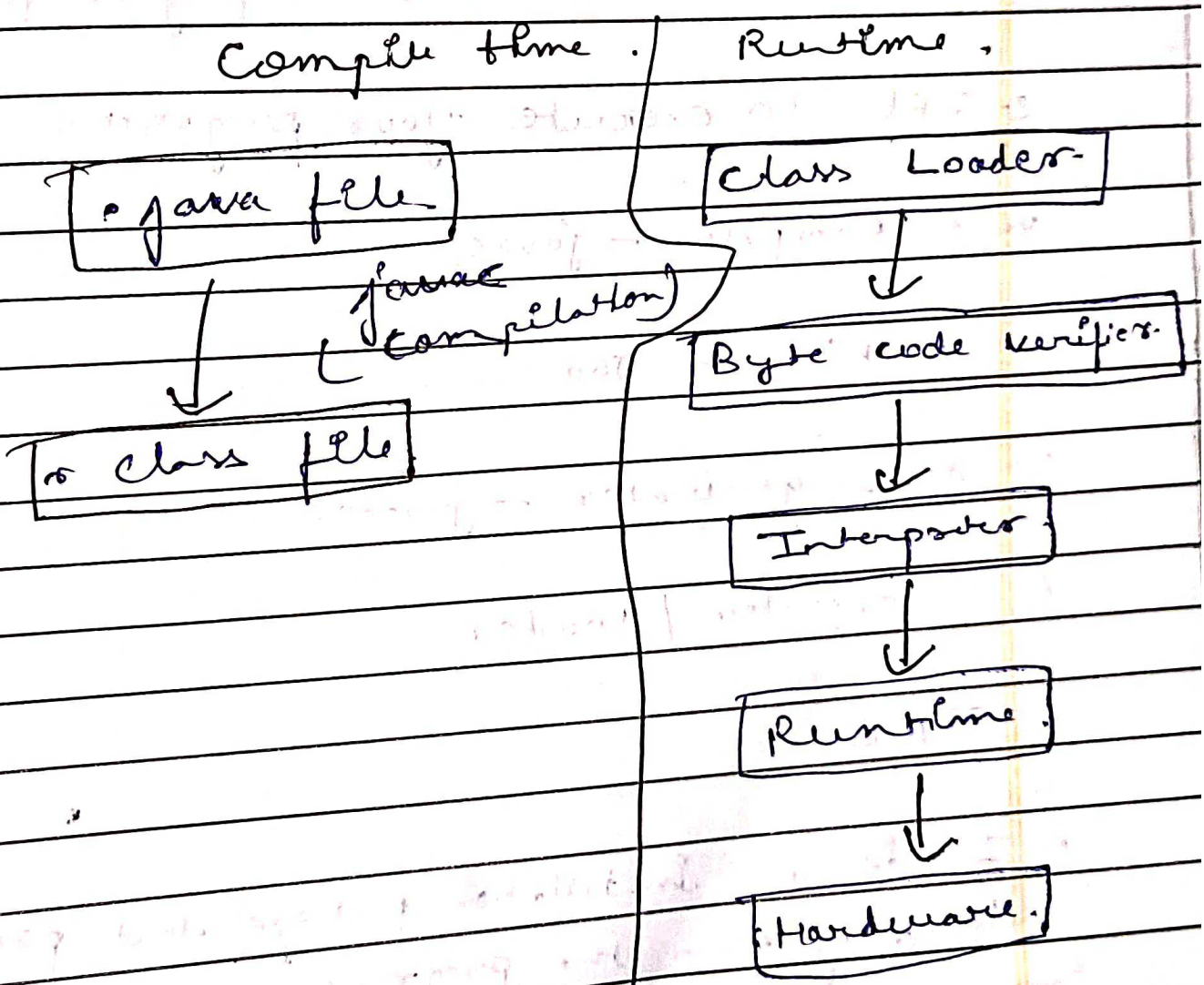
JRE

- It is an Installation package that provides environment to only run the program
- It consist of : 1. Deployment technologies
  2. User Interface toolkits
  3. Integration libraries
  4. Base Libraries
  5. JVM.

• After we get the .class file, the next things happen at runtime:

1- class loader loads all classes needed to execute the program.

2- JVM sends code to Byte code verifier to check the format of code.



(How JVM works) Class Loader.

- Loading:
  - class reads .class file and generate binary data
  - an object of this class is created in Heap.

- Linking:
  - JVM verifies the .class file.
  - allocates memory for class variables & default values.
  - replace symbolic references from the type with direct references.

- Initialization:

All static variables are assigned with their values defined in the code and static block.

JVM contains the stack and Heap memory allocations.

JVM execution.

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

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

Garbage collector.

DATE: / /