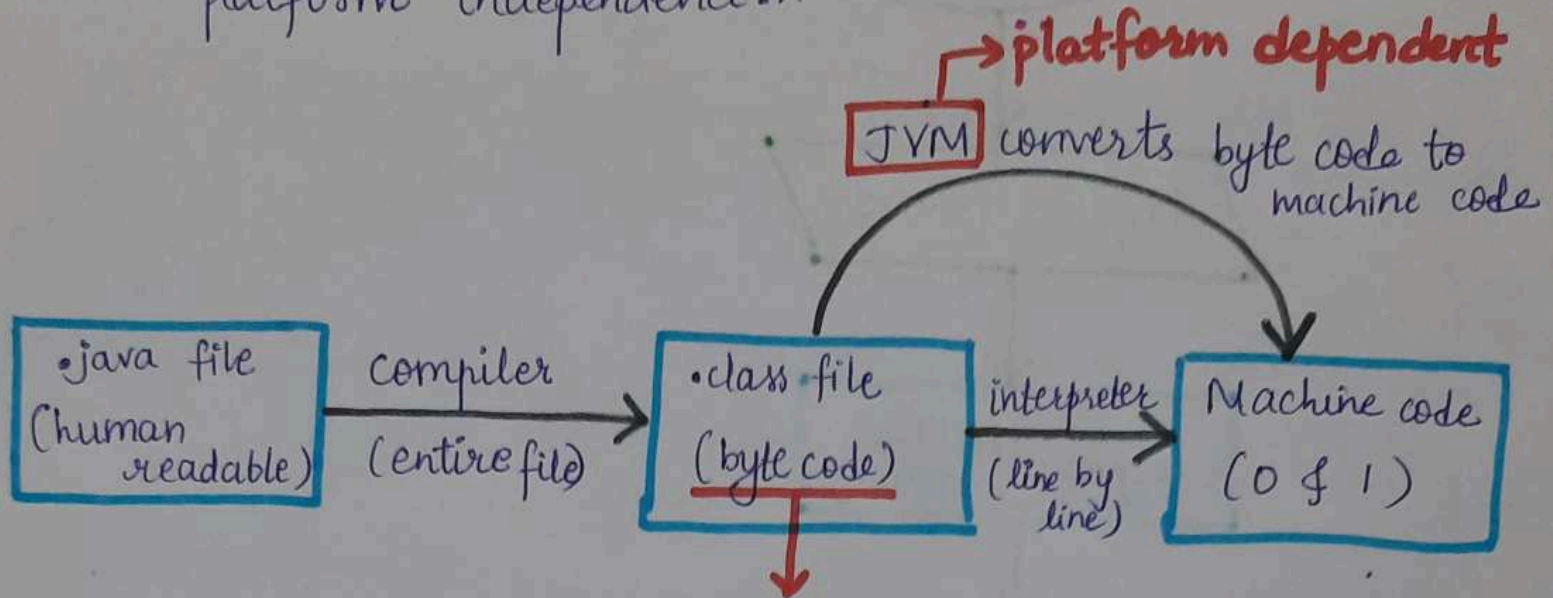


3/8/21

Introduction to Java ♥

★ How Java code executes and more information about platform independence...



- can run on all O.S.
- this code doesn't run directly on a system, for this we need JVM

★ 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 executable file that we get, it has step by step set of instruction dependent on platform.

★ JDK vs JRE vs JVM vs JIT

JDK [Java Development Kit]

↳ provides environment to develop & run Java program

JRE [Java Runtime Environment]

↳ provides environment to only run the program

JVM [Java Virtual Machine]

JIT
~~[Java]~~
[Just-in-time]

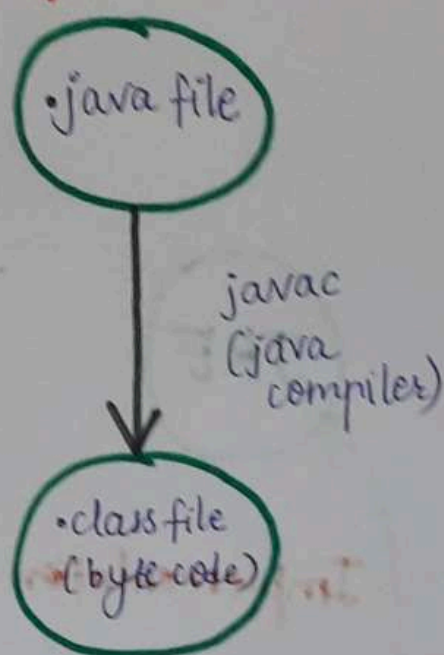
→ Java Interpreter
→ Garbage collector
etc.

→ deployment technologies
→ user interface toolkit
→ integration libraries
→ base libraries
etc.

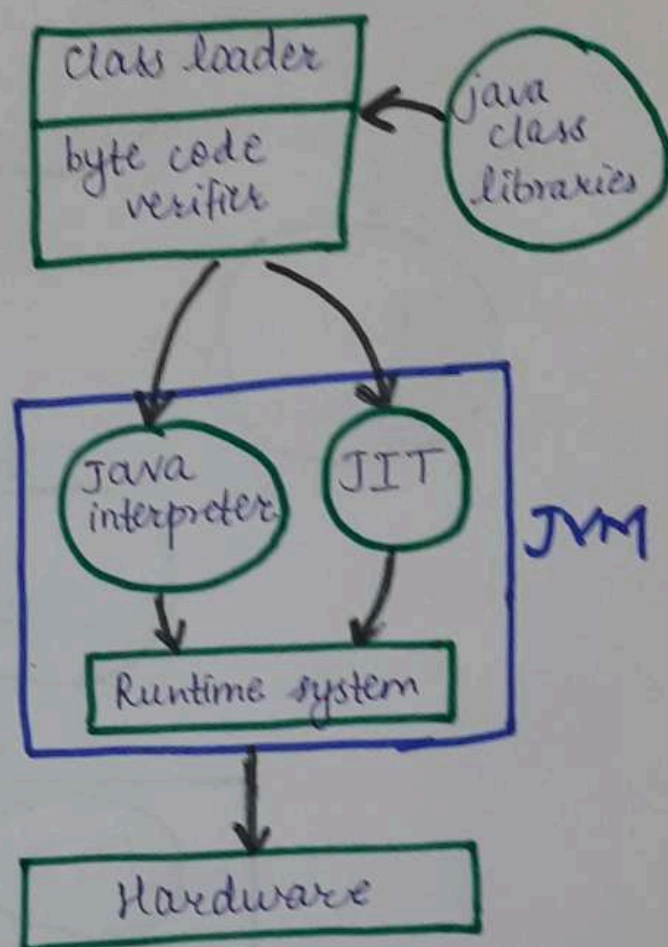
→ development tools
→ javac → Java compiler
→ archiver → jar
→ docs generator
↳ javadoc
→ interpreter/loader
etc.

Java Development and Runtime Environment

Compile time



Runtime



⇒ JVM execution:

• Java Interpreter:

- line by line execution
- when one method is called many times, it will interpret again & again

• JIT:

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

• Garbage Collector

★ Class loader:

• Loading

- reads byte code file & generates binary data
- an object of this class is created in heap

• Linking

- JVM verifies .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 & static block.

★ Summary:

