Introduction to Java :-

Java is a programming language.
↳ It is object-oriented programming language.
↳ It is developed by Sun Microsystems in 1995.
↳ Now it is subsidiary of Oracle.

Java is currently used in development of :

- Web applications
- Desktop "
- Enterprise "
- Mobile
- Games, etc..

Types of Java Applications

Standalone
↳ Desktop applicats.
Ex: Mediaplayer
↳ AWT & Swing are used to create these application. (in Java)

Web
↳ Runs on browsers.
↳ Uses JSP, Servlet, etc..

Enterprise
↳ Banking applicats
↳ High-level security
EJB is used

Mobile
↳ App for mobile devices

Types of Editions

Java SE
↓
Standard Edition
↳ programing platform
↳ Includes lang, util, io, math etc. packages & OOPS, exception, AWT, Swing etc..

Java EE
↓
Enterprise Edition
↳ Enterprise platform for web & enterprise application.
↳ Includes Servlets, JSP, Webservices, EJB etc..

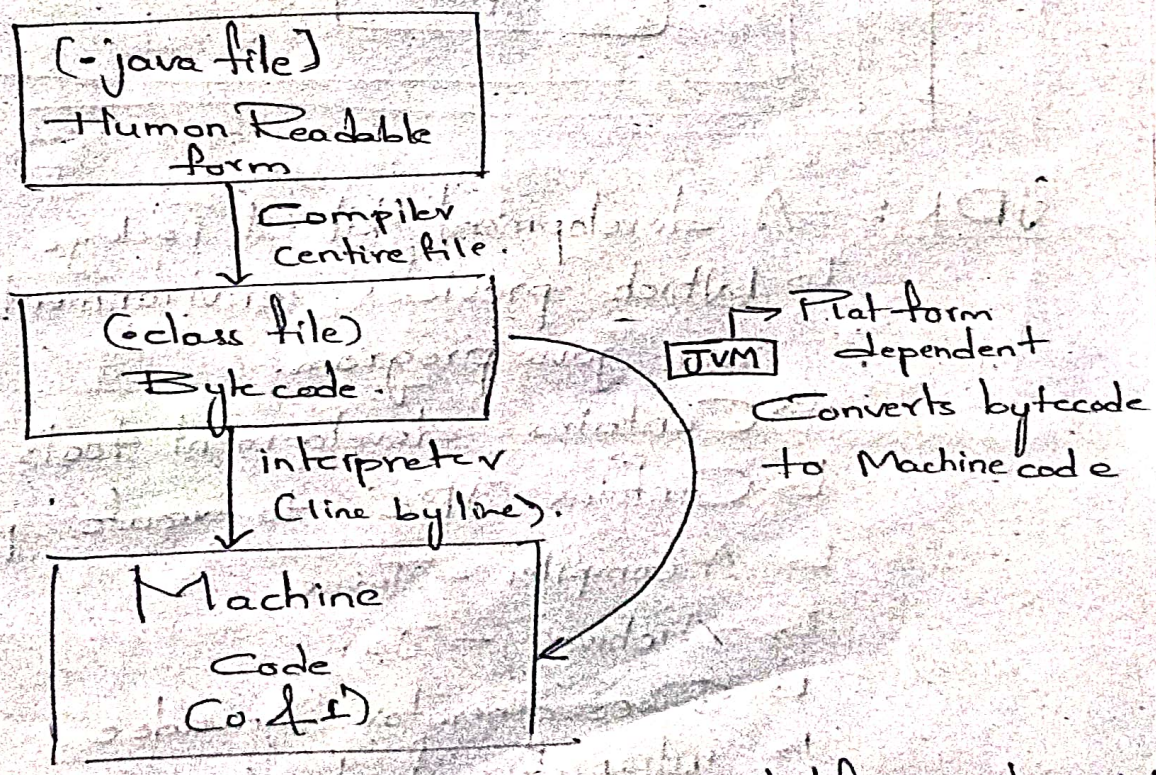
Java ME
↓
Micro Edition.
↳ for mobile applications

Java FX
↳ to develop rich internet applications

Features of Java:

- Platform Independent
- Object-Oriented.
- Secured
- Robust.
- Multithreaded
- Distributed
- Dynamic.
- High Performance.
- Simple.

Explanation for why java is platform Independent & Execution of java code:

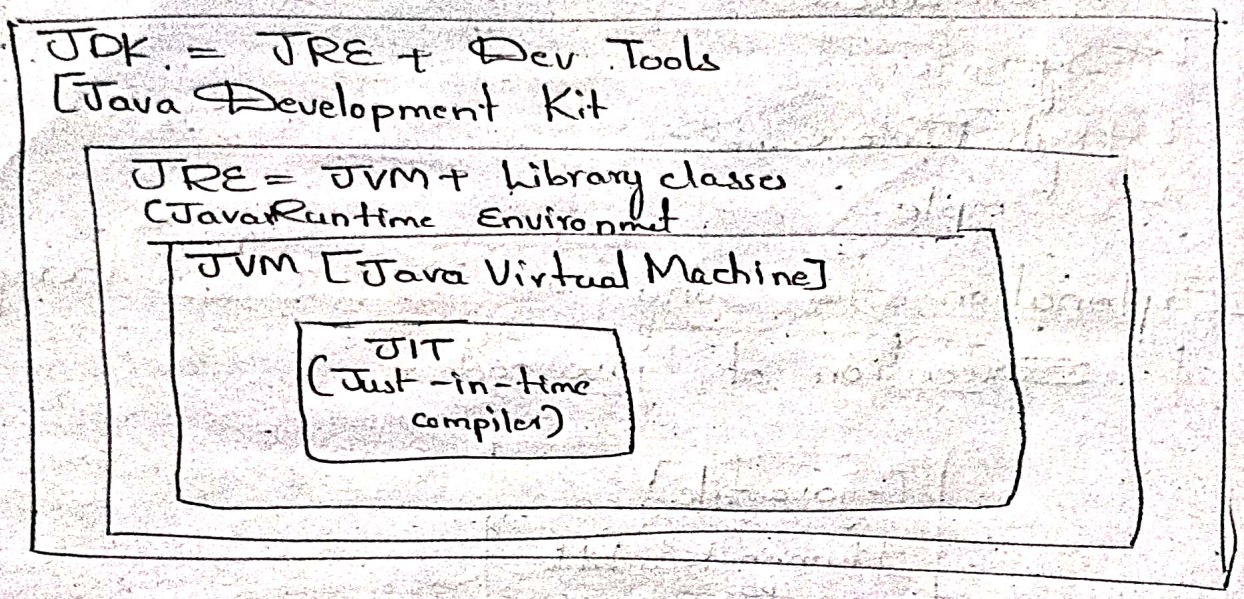


Byte Code: It makes the Java platform independent

- This byte code can be run on all Operating Systems.
- But it doesn't directly run on a system. It need JVM to do that.

↳ Where as JVM is platform-dependent
ie; for every Operating System the executable file we get, it has step-by-step set of instruction dependent on platform.

Architecture of Java.



JDK :- A development kit & Package which includes.
- ↳ Which provides environment to develop & run java program.
- ↳ Contains development tools.
- ↳ Contains JRE - to execute program.
- ↳ A compiler - Javac
- ↳ Archiver - Jar
- ↳ Docs generator - javadoc
- ↳ interpreter & loader.

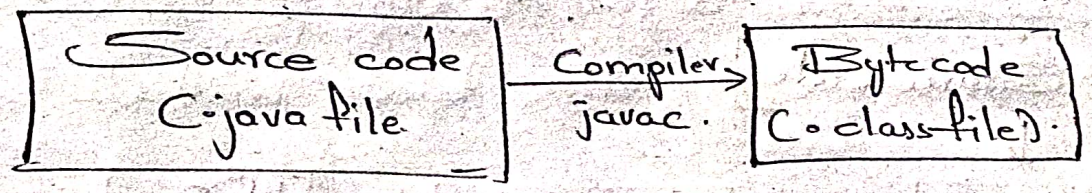
JRE: It is an installation package which provides environment to run the java program.

↳ It contains
- ↳ Deploy-ment technology
- ↳ UI toolkit
- ↳ Integration libraries & Base library.
- ↳ JVM.

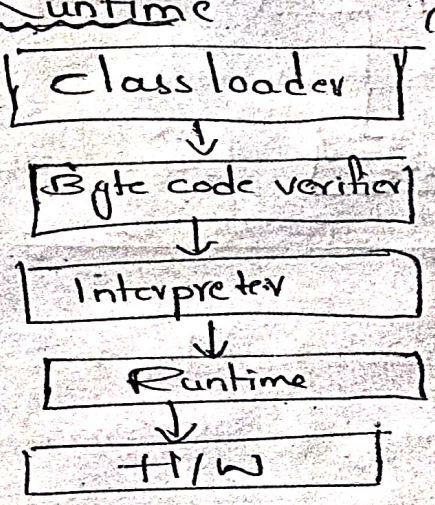
JVM

Compile-time & Runtime scenarios of a java program:-

Compile time



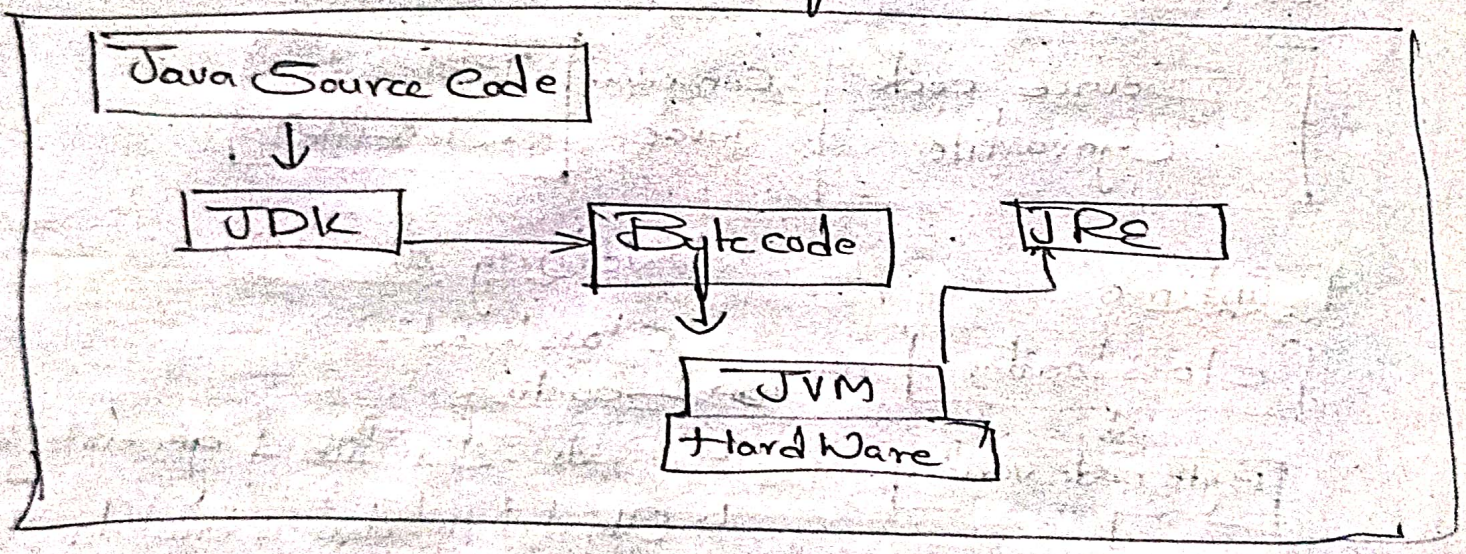
Runtime



→ Working of JVM

Class Loader

- Loading
  - ↳ Reads .class file & generate binary data.
  - ↳ Object of this class is created in heap
- Linking
  - ↳ JVM verifies .class file
  - ↳ Allocates memory for all class variables
  - ↳ Replaces symbolic references with direct
- Initialization
  - ↳ All static values are assigned
  - ↳ JVM contains stack & heap memory location.

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 that interpretation is not required.
  - ↳ Makes execution faster.
  - ↳ Contains Garbage collector.