

# ☑ Roadmap of course And ☑ Fundamental of JAVA

classmate

Date 07/10/22  
Page

⇒ ROAD MAP :-

core java = Pojectee + JEE & SpringBoot

(i) Fundamentals (Basic)

• Everyweek Assignment = (2)

(ii) Git & Github (6hr)

• Every month Quiz =

(iii) Fundamentals of java (OOP, loop, DT, Array, String ...)

(iv) OOPS (inheritance, Polymorphism, Int, Abs,

(v) exception handling, Multi threading, Collection, Annotation, enum, file handling, String API, Lambda exp. ...

(vi) SQL → MySQL (vii) NoSQL → MongoDB, (viii) JDBC

(ix) HTML & CSS → web project. (x) → 2.5hr

(xi) JEE (Servlet & JSP) → 2 week

(xii) JavaScript & React.js, (xiii) Hibernate

(xiv) Spring core, Spring Boot, ...

(xv) Docker, Kubernetes PLF

(xvi) Microservices (AM)

(xvii) Agile & Scrum } → capstone project (Java + React.js)

## ★ Fundamental of JAVA ★

- Before java there was two language who are ruling the industry → C, C++ by Bellab
- 1991 → the project of java was started by the startup named Sun Microsystems (was a set-top box, TV sets) with many developer. And lead of the project was JAMES GOSLING. one of the co-founder was Vinod who was Indian
- Reason of java success ⇒ Easy to understand, object-oriented and portable/platform independent/WORA (Write once Run Anywhere). C and C++ were not portable.

I.Q. • 1995 ⇒ Trial version released (Alpha & Beta version) and they made it freely downloadable, open source.

I.Q. • Jan 1996 ⇒ officially released named as Oak (Java 1.0)

• Name :- Java (coffee plant in Indonesia).

• Previous decided name which failed :- oak, green, c+++, ...

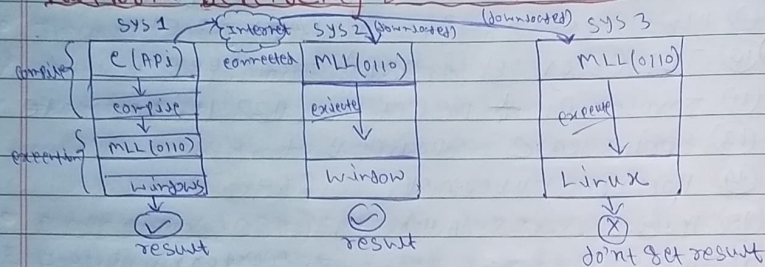
• 1995 to 2011 ⇒ Java was the top language even today.

I.Q. • 2011 ⇒ Oracle acquired java from Sun Microsystems and made it closed source.



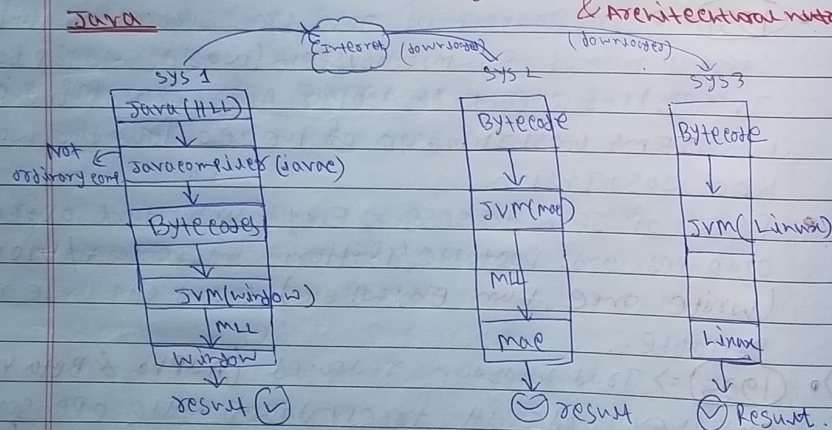
- 1.0) Feature of java:- (i) Platform independent or Portable or WORA, (ii) Object oriented, (iii) Secure, (iv) Robust, (v) Internet Portable, (vi) Multitasking, (vii) Distributed, (viii) Dynamic, (ix) Simple, (x) Interpreted and High performance.
- Platform:- IS a combination of MP + operating system the one who interact/operate with MP.
- ex: intel + window (MP) (OS), ~~mac~~ M1/M2 + mac (MP) (OS), intel + linux (MP) (OS)

### Platform Dependency & non-Portable:-



So, MLL says if it compiled in window it will work only on windows. So, C++ makes Platform dependent.

### 1.1) How Platform Independent & Portable & WORA? :-

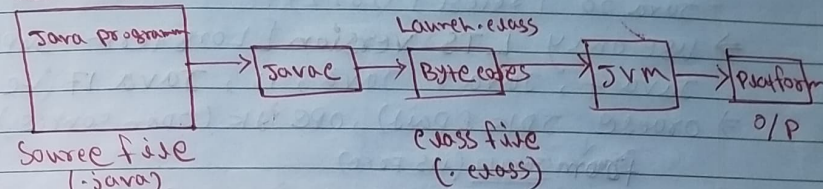


- What is Bytecode:- A special compiler in Java convert HLL to a Intermediate language (which is neither HLL nor MLL) This is called Bytecode. Bytecode Portable But JVM only PLATFORM DEPENDENT. This why we need to download

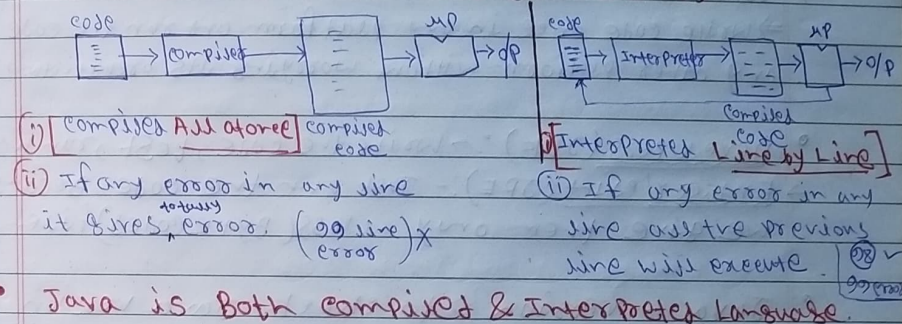
diff. diff. JDK version of diff. diff. platform to execute Bytecode.

- Program / application / Software  $\rightarrow$  all are same.

### Launch.java



### Compiler vs Interpreter (Difference):-



### Java is Both compiled & Interpreted Language.

### All the version of JAVA:-

- C  $\rightarrow$  1970 (1969-1973) by Dennis M. Ritchie
- C++  $\rightarrow$  1983 by Bjarne Stroustrup
- Python  $\rightarrow$  1989 by Guido van Rossum
- Java  $\rightarrow$  1995 (Trial version Alpha & Beta)
- 1996  $\rightarrow$  officially Java 1.0 (oak) [JDK 1.0]
- 1997  $\rightarrow$  Java 1.1 [JDK 1.1]
- 1998  $\rightarrow$  Java 1.2 [JDK 1.2] - collection framework, - Swing, - JIT compiler
- 2000  $\rightarrow$  J2SE 1.3
- 2002  $\rightarrow$  J2SE 1.4
- 2004  $\rightarrow$  J2SE 5.0 / JAVASE 5 [Annotations, Auto boxing, Enumerations, Enhanced for loop]
- 2006  $\rightarrow$  Java SE 6
- 2011  $\rightarrow$  JAVASE 7 [string in switch, - try with resources, < > dimo]
- 2014  $\rightarrow$  JAVASE 8 [Lambda exp, support for JS code, Data and Time API]
- 2017  $\rightarrow$  Java 9 [- modularity, - Reactive stream, JShell]
- 2018  $\rightarrow$  Java 10 [- local variable type infer]



sep. Java 11 → 2018 → [- Run Source file, - var for lambda  
Java 12  
Java 13 - sep 2019 → [- Switch Expression, - multiline strings  
Java 14, Java 15, Java 16, Java 17, Java 18, Java 19  
Java 20 [JDK 20] (2022)

→ Use only **LTS version** [Long term support]  
Java 7, Java 8, Java 11, **Java 17** (we use it)

→ Oracle JDK (paid), OPE JDK (open source), Amazon Corretto  
from Java 17 (its free)

• Java provide backward version capability to run old versions.

Q) What is architectural neutral?

→ Write code once in any platform and run the code in any other platform without worrying its architecture neutral.

• Developers (JDK) :- write the code, test, run the code.

→ End user / client (JRE) :- Just run the code. Because we give them only .class (Byte codes) file.

• **Java --version** → to check Java version

I.Q) Can we see file containing machine code?  
Ans:- yes. (for c, c++)

Q) Can we see Bytecode's machine code?

Ans - NO, Because output file <sup>(final)</sup> is not generated in Java.

I.Q) What is jar file and war file?

→ ~~Java~~ Jar file → zipping / compressed of so many .class file. It stands for **Java Archive file**.

→ War file → web application resource / web Application Archive. It contains JAR, Java server pages, Java servlets, Java classes, XML, tag libraries, static site (HTML).