→ Enhanced Version of HDD is SDD.

→ SSD → Solid-state drive.

→ Semiconductor Chip Shortage.

19/10/22 ② Roadmap of Course And fundamental of Java :-

Radmap
video, notes, Assignment
Technical discussion

doubt Session

wed : 2pm to 4 pm IST

Sat : 6Pm to 8Pm IST

3Pm to 10Pm mon-fri
Chat support

→ 3h x5 = 15h ⇒ 10+2 → waklig

1h → Practice.

→ Core-java ⇒ practice it on IDE

① fundamentals (Basics)
② Git & GitHub } 6h ⇒
③ fundamentals of Java
↳ loops, String, Array.

④ OOPs (Encapsulation, poly, inhe)
&
⑤ Exceptional handling. Etc.
Core Java → 1½ Month

Every week
Assignment - 2
2 Assignments
Module - quiz.

⑥ SQL → MySQl
No SQL, MongoDB.
↳ 1 week

⑦ JDBC → 3 days

~~⑧ JSP~~

→ HTML & CSS
↳ self Paced

⑧ JEE (Servelets & JSP)
↳ 2 week / 1.5 Min max

⑨ Javascript + React js

⑩ Spring Core

⑪ Spring Boot, JPA, REST

⑫ Docker, Kubernetes.

⑬ Microservies.
AWS

⑭ Agile & Scrum
↳ 3 Projects.

Attend

①live Classes | Recording

Java

19/10/22

1970's 1980's

→ Bell Labs (C & C++)

↳ before Java these P.L ruled.

↳ now its Turbo C

① Java Introduced in 1991

↳ Sun Microsystem-Company.

↳ James hosting

↳ Standford univerty students.

→ Easy to understand

→ Object oreinted

→ Portable /Platform

→ platform Independent

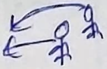
} ~~wora~~ WORA

→ C & C++ Programs are not portable and platform dependent.

→ In 1995 → Program is ready

↳ Alpha & Beta Version

↳ free down loadable.

↳ open sourse. 

] → any one can contribute

↳ James hosting.

→ 1996 → they released Java 1.0 → oak.

↳ it is platform Independent and Portable.

↳ it is architecture neutral. (or Internet Programming language

↳ object oreinted.

→ Naming this programming language

↳ oak, Green team,

11 member Alphabet.

↳ Coffee planter island located in Indonesia. → Java

→ In 2011 → oracle acquired Sun Microsystem.

→ Java ⇒ 1995 - ~~Java 1.0~~ ⇒

Java 1.0 ⇒ 1996.

→ it is object oreinted.
platform Independent
(or)
Portable
(or)
WORA

→ Internet Programming language.

→ it is distributed. and secure.

20/10/22

Portable / Platform independency / WORA

① Platform:- Processor + OS → Called platform.

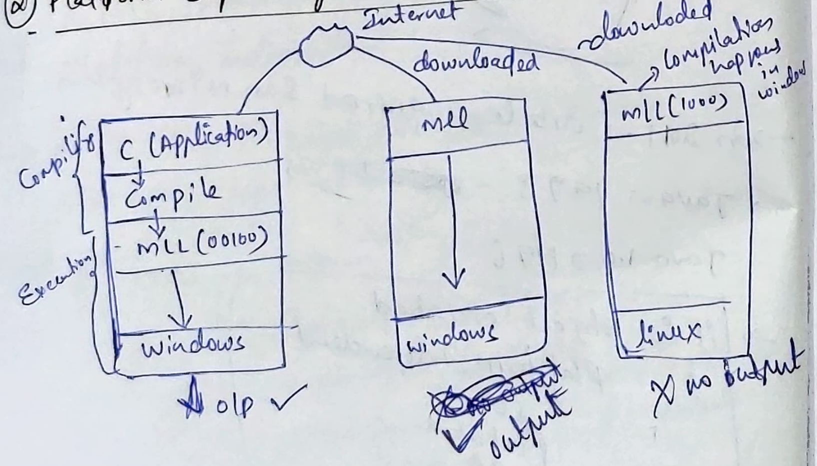
intel + Windows

M1 ~~[illegible]~~ + Mac

intel + linux

S/W ⇒ platfrom ⇒ OS → windows
mac
linux

② Platform dependency & non-portable :-



→ Program (or) Application (or) software refers Same.

Case①:-

platform of	Compilation	Execution	Result
	windows	windows	✓ we get o/p

→ over network act Code never be transported.

→ only Compiled Code will be transported. (MLL)

Case	Compilation	Execution	Result.
①	windows	windows	✓ (we get result)
②	windows	windows	✓
③	windows	linux	X (no output)

→ MLL is always platform dependent.
↳ if it is Compiled in windows then it Executes in windows system only.

(i) Platform dependency:- if a programming language is platform dependent Such a P.L where platform of Compilation and platform of Execution has to be Same.

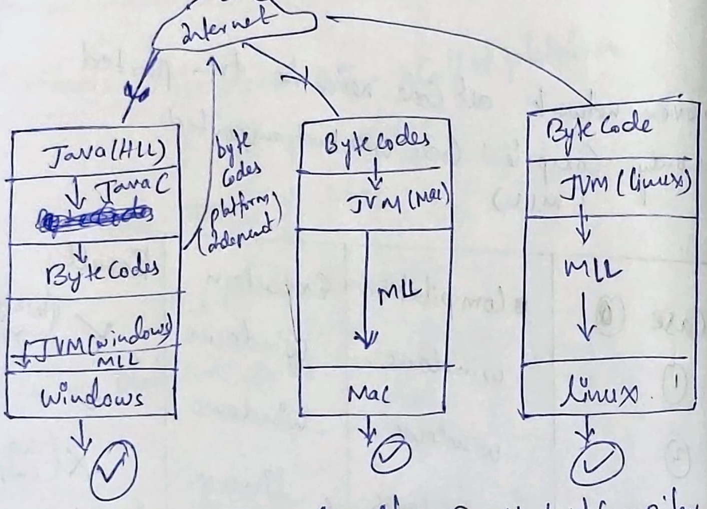
Eg:- C & C++ platform dependent program.

→ we Cannot Port it and Execute it. hence it is called non-Portable.

Java

↳ ~~How~~ java ~~achived~~ ~~independency~~ ?

How Java achived independency ?



→ Java used it own Compiler (or) Hybrid Compiler

→ Java Compiler (or) Java C

→ to make Java portable (or) platform Independent

→ ordinary Compiler Converts HLL to MLL directly

→ to make Jave portable (or platform Independenky (or) WORA theya are Java C (or) Special Compiler.

→ This Complier ~~C~~ will not Convert Java to M.LL rather it Converts to Byte Code.

(i) Byte Code :-

→ Intermediate level code

→ they are neither in H.L nor in M.L.

→ When Java Code is Complier It is Converted to byte Code.

→ Byte Codes are Platform Independent,

↳ it works on any Platform.

→ JDK → JRE, Javac, JVM

→ JDK is different for all Platforms.

↳ JVM is platform dependent.

↳ different JVM's for all platform.

↳ windows, linux, macintosh.

JDK ⇒ windows, linux, macintosh.
JVM ↗

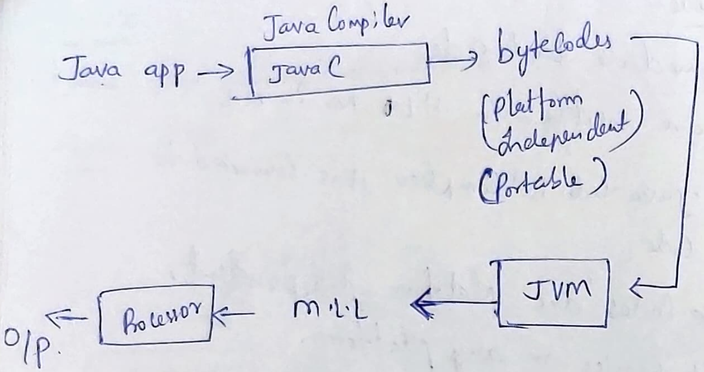
(ii) JVM :- designed using c language..

JVM is platform dependent.

→ JVM is different for different platform

→ JVM ~~Converts~~ will Convert byte Code into M.L.L (or) Machine understand language by Internally makes use of Intrepreter. & takes Care further Execution in Java.

→ Write once & run anywhere.

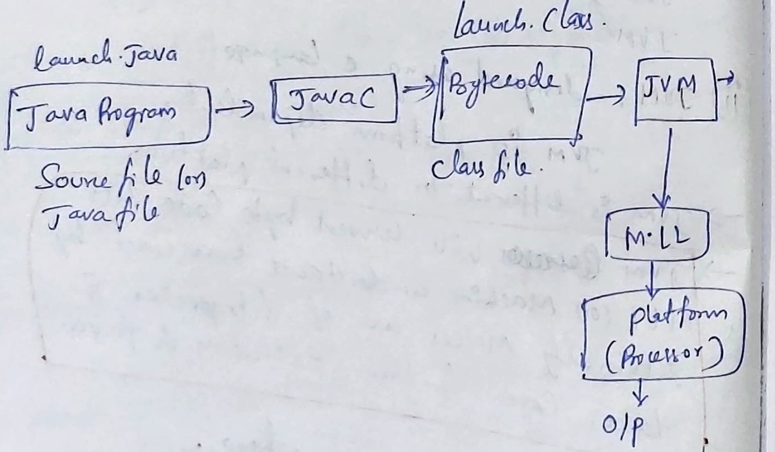


→ Java is architectural neutral.

→ internet programming language

→ WORA :- write once and Run Anywhere

→ it take More time for Execution than C++ & C.

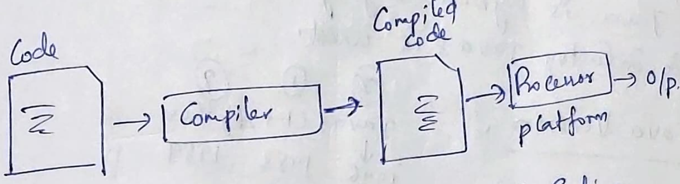


(iii) Class file :- the place where byte codes are Present is called class file.

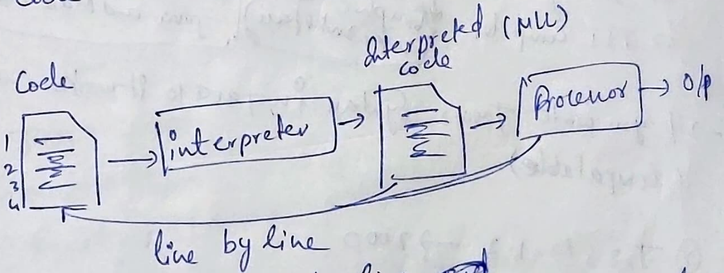
→ .class (Extension)

→ After Compiling Java program it will genrate Class file (byte codes).

(iv) Compiler vs Interpreter :-



→ Compiler will translate (or) Compiled Entire Code all at once.



→ it Interprets line by line.

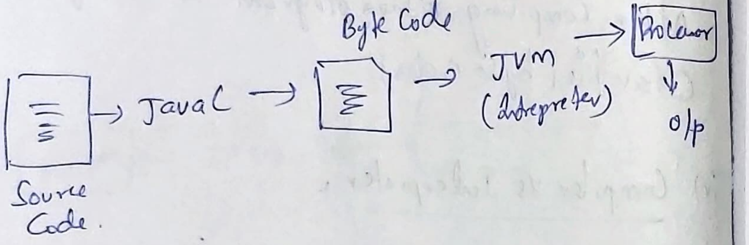
→ if there mistake in 99th line Compiler throughs Error. In Interpreter 98 line will work.

→ Eg:- Python, Java Script all are Interpreted Programming Language.

→ Jave is Compiled and Interpreted language

→ Compiled at Early stage.

→ Interpreted at Execution stage.



→ Java is hybrid language.

→ In Industry Java 8 is used.

③ ① ②
Java, C++, Python
↓
1996 1983 1989
1998→J2SE 1.2.

Java Version :-

① J2SE 1.2 – 1998
↳ Collection framework
↳ Swing → GUI ↓ Graphic user Interface
↳ JIT Compiler

C++ → C#

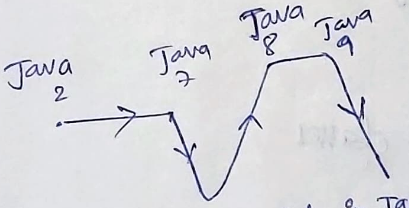
Microsoft Copied from Java and Mad C#

→ if you write Java 2 Syntax in Java 10 it works (Compatable).

② J2SE 1.3 → 2000
③ J2SE 1.4 → 2002
④ J2SE 5.0 → 2004
↳ Annotations
↳ Auto boxing
↳ Enumeration
↳ Enhanced for Loop.

⑤ Java SE6 – 2006
⑥ Java SE 7 → 2011 → ~~bech~~ bench Mark
oracle acquired
↳ String in Switch
↳ try with resource
↳ < > diamond ~~operator~~ operator.

⑦ Java SE8 – 2014 bench Mark.
↳ lambada Expression
↳ Support for JS Code.
↳ Date and Time API
↳ Stream API.



→ we Can write JS Code in Java.

⑧ Java 9 – 2017
↳ Modularity
↳ Reactive streams
↳ JShell
→ oracle Mad JDK paid. (for Commericial)
→ licence fee
→ cloud started famous.

(9) Java 10 - 2018.

↳ local variable Type Infe

(10) Java 11 - Run source file

↳ In one step we can run Java Code

→ Var for lambda.

(11) Java 12 -

(12) Java 13 - Switch Expression

Multiline-Strings.

(13) Java 14 - Records

- Packaging tool

(14) Java 15 -

(15) Java 16 -

(16) Java 17 - sealed classes

(17) Java 18 -

(18) Java 19 - Virtual threads

- vector API.

→ 70% of Industry project are Java 8.

LTS Versions Long term Support.

↳ Provided by Company

↳ Java 7
↳ Java 8 ✓
↳ Java 11 ✓ focus] Company using this Version.
↳ Java 17 ✓

→ oracle JDK Paid

↳ open source JDK (Alter).

↳ Amazon Corretto. ✓

→ Oracle JDK free from Java (17)

→ Jar → Zipping of .class files is Called as ".jar" file.

→ it stands for Java Archieve file