Introduction to Java

lec. 2

A Brief History: Story behind Java's Creation

- Sun MicroSystems was a well-known American company that sold computers, computer components, software & information technology devices. But it got acquired by Oracle in 2010.
- In 1991 the company wanted to check what's next in computing? Hence the story of Java begins in June 1991
- So they assembled a team named as "The Green Team" to work or to brainstorm on this idea. So this Team decided to do something for electronic devices, home appliances (TV, setup box, etc). So they named this project as "Green Project"

A network of heterogeneous consumer devices communicating with each other could be a trend.

- The vision for this project was to build an interactive wireless handheld device which would communicate with home entertainment devices like TV, VCR and the software that Green Team develops would be installed on all of these devices. (It's like a network of heterogeneous devices)

* But there were some challenges in achieving this goal..
  - → As the devices (TVs, VCR) have less limited memory. So the software should consume less memory
  - → There devices has their own Hardware / Platform. So the software ~~should~~ Program should be platform independent So that it can run on any device
  - → Security (There software should not cause any harm to the target devices)

multi-threading

- Initially C++ was considered for this project but it was not a good fit for this.
- So The team which was led by "James Gosling", Mike Sheridan and Patrick Naughton decided to create a new technology / language.
- One of the team members, James Gosling created an entirely new language and named it as Oak, after the tree outside his office

- So James Gosling was known as Father of Java

1991 - 1994 :-

The Star7 device was one of the first projects where Oak was implemented.

(In 1992) → The Star7 was designed as a type of PDA (Personal Digital Assistant) / smart device that could interact with users through a graphical interface

But Cable Tv industry was not interested in this because this technology was far too advanced for them

But then in mid 1990s internet was rapidly growing So the Green Team recognized that Oak's Platform independence could solve a much larger problem : building softwares that could run on different computers connected via web (www).

Their focus shifted from TV devices to web applications

1995 :- Oak was renamed as Java due to some trademark issues

Q. Why they name it Java?

- Because of Java Coffee which was popular among the team members during that time of development.

[Java is an island of Indonesia where coffee was cultivated on a large scale by the Dutch during 17th century]

In 1995, Time Magazine called Java one of the Ten Best products of 1995

In 1995 Java was officialy launched. and during that time Java meets WWW. before Java web pages were mostly static (using HTML) So Sun developed their first web browser HotJava to showcase the power of Java Programming language.
And this browser was designed to run Java applets (small, interactive programs that could be embedded in web pages to make them dynamic. and interactive

HTML + Java applets = dynamic → How it works? See on last page

NOTE:- Applets have been removed from modern versions of Java because of security issues, declining usage, discontinued browser support. As Now a days HTML5, CSS, JS is being used to make interactive web pages.

But HotJava never became popular browser. Eventually other more successful browsers like Netscape Navigator and Internet Explorer adopted support for Java applets

As the Internet exploded, So did Java's Adoption By 1996. the first official release of Java (JDK 1.0) was out and it quickly became go-to language for developing web applications, enterprise software, desktop applications & mobile applications

NOTE:- Java's cross-platform capabilities helped it become one the most widely used programming language in the world.

Latest version of Java = Java 23

LTS versions:- Java 8, Java 11, Java 17, Java 21

Versions of Java :-

250 classes ← Java SE 1.0 (1996) - slow
500 classes ← Java SE 1.1 (1997) - little faster
more than 1500 classes ← Java SE 1.2 (1998) - much faster
Java SE 1.3 (2000)
Java SE 1.4 (2002)
2500 classes + generics + multithreading ← Java SE 5 (2004)
Java SE 6 (2006)
Java SE 7 (2011)
Java SE 8 (2014)
Java SE 9 (2017)
Java SE 10 (2018)
Java SE 11 (2018)
Java SE 12 (2019)
Java SE 13 (2019)
Java SE 14 (2020)
Java SE 15 (2020)
Java SE 16 (2021)
Java SE 17 (2021)
Java SE 18 (2022)
Java SE 19 (2022)
Java SE 20 (2023)
Java SE 21 (2023)
Java SE 22 (2024)
Java SE 23 (2024)

⇒ Java technology is both a programming language and a platform.

Java PL is a high-level, object-oriented language that has a particular syntax & style.

A Java Platform is a particular environment in which Java programming language applications run.

- Java Platform, Standard Edition
- Java Platform, Enterprise Edition
- Java Platform, Micro Edition
- Java FX.

Java SE is used for developing general purpose applications ③
→ It defines everything from basic types and objects to high-level classes that are used for networking, security, database access, GUI development etc.

Features of Java -

① Platform Independent :-
Java applications can run on any operating system or Hardware without modifications provided there is a compatible Java Virtual Machine available for that platform.
Java follows WORA (Write Once Run Anywhere) ~~Read~~ principle means once you write and compile a Java program to bytecode, it can run on any platform using platform's JVM.
This makes Java highly portable across operating systems

② Simple :- removes many complex features of C & C++ like:
- pointers
- multiple inheritance

③ Secure :- Java applications can be downloaded from anywhere but
→ it make sure it doesn't harm to your computer as the bytecode of that application runs on JVM (Java virtual machine) JVM provides an isolated & restricted environment for running programs.
→ Security manager is a key feature in Java that defines access control rules for Java applications running in a restricted environment often referred to as a sandbox.
→ A sandbox is a secure & isolated environment where code can be executed safely without affecting the host system or other applications.

eg:- modern web browsers use sandboxing to run web applications and scripts
Mobile applications run in their own sandboxes, one app can't interfere with another.
Docker allow developers to create sandboxed environments for applications, ensuring that dependencies & configurations don't affect other projects.

(8) Bytecode verification :- In Java programs are compiled into bytecode, which is then executed by the Java Virtual Machine (JVM). Before execution, JVM verifies the bytecode to ensure it adheres to Java's security rule

(9) Robust :- It is robust and we can say reliable because of the following properties:

(i) Automatic Memory management / Strong Memory Management
↳ It provides automatic garbage collection which helps manage memory & reduce the likelihood of memory leaks & pointers errors. No need to add memory management logic.

(ii) Strong Type checking :- Java is statically typed, which means type checking is performed at compile time. So this helps to catch errors early in development process

(iii) Exception handling :- exception handling means managing errors that occur during a program's execution so that the program can continue running smoothly instead of crashing
Java has a strong exception handling and because of this the program can handle errors without crashing, providing meaningful feedback to the user.

eg:- ClassNotFoundException IOException etc.

5) Object-Oriented Progeamming :-

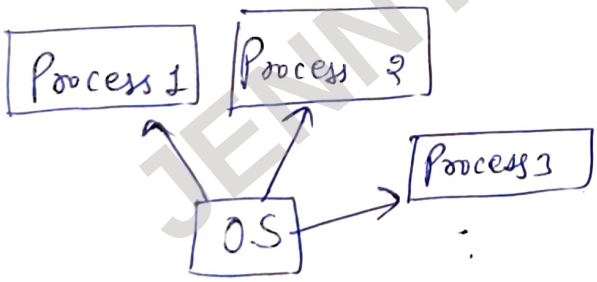
Java supports OOP features (Abstraction, Encapsulation, Inheritance & Polymorphism)
Java's Object Orientedness helps model real-world scenarios in a more natural way.

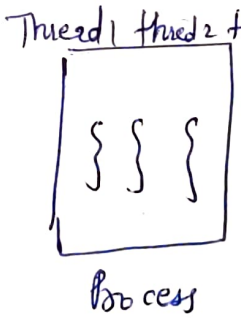
(6) Architectural Neutral:-

- Java Compiler generates bytecode that is architecture-neutral, meaning it can run on any machine that has the JVM installed, regardless of the underlying hardware

(7) Multithreaded:- Java has built-in support for multithreading, allowing concurrent execution of two or more parts of a program for maximum CPU utilization.

eg:- in a restaurant multiple waiters (threads) are serving customers (tasks). Each waiter can take & serve orders independently without waiting for others to finish. So they are able to serve customers faster & more efficiently.





eg:- A word processor can have multiple threads running like:- one thread running in foreground as an editor & another in the background auto saving the document, one thread to check the errors while you working.

So thread is a unit of a process.

- Multithreading is used in web servers to handle multiple client requests at the same time
- in Games, separate threads can handle rendering graphics, processing user input, play audio etc.

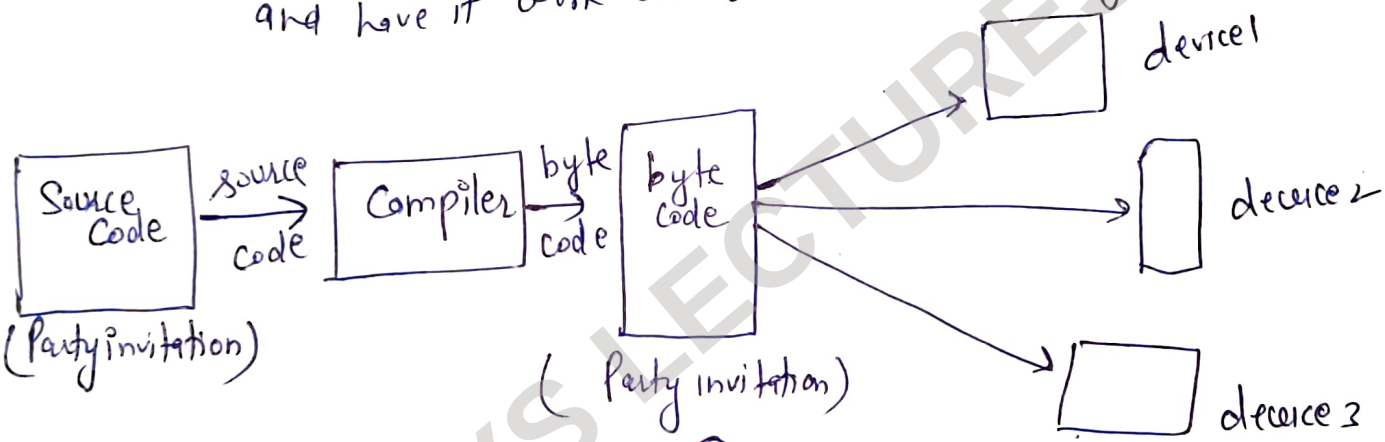
What is Java?

It is a general-purpose, Object-oriented, high-level language which is platform-independent and very fast.

It is widely used for developing applications across various platforms such as web, mobile, desktop and embedded systems

How Java Works?

e.g. - The goal is to make/write one application (invitation) and have it work on whatever device your friends have.



① Java Program [.java file]

② Compiler checks for errors & generate an intermediate code (byte code) [.class file]

③ This compiled bytecode is platform-independent

④ All these devices should have JVMs to run this byte code.

NOTE:- One more imp. feature of Java is Backward Compatibility. That means newer versions of Java language & JVM can run applications written in older versions of Java without requiring modifications. It is one of the reasons of its long-term popularity. Developers can confidently upgrade to newer versions of Java, knowing that their existing code will work continue to function as expected.

Who is using Java:-

① Big tech companies: Google, LinkedIn, Netflix, amazon, gmail

② Java was used by NASA for their Mars Rover project called spirit

③ Open source libraries :- most open source libraries are implemented in Java.
eg:- Apache Solr, hadoop, etc.

What is Java SE ?

- As we know we can create Java applications for different devices like desktops, web servers, mobile devices & so on and there are separate platforms dedicated to develop different type of applications.

So here we will discuss Java SE

Editions / Platforms of Java software family :- mainly 3

① Java Standard Edition (previously called as J2SE)
↳ used to develop standalone applications that typically run on desktops & servers
eg:- Hospital Management System, used within a Hospital
or Hotel Management System.

② Java Enterprise Edition (previous called as J2EE)
↳ used to develop large scale applications runs on servers.
e.g:- e-commerce website.
large no. of users are accessing it at any particular time.

JSPs, ~~Servelts~~ Servlets

- Java EE built on Java SE
- Java EE has been rebranded as Jakarta EE bec
  It has been shifted to Eclipse foundation.

③ Java Micro Edition: (previously known as J2ME)

↳ used to develop applications for mobile, set-up boxes (resource constraint devices)

→ subset of Java ME.

NOTE:- You need to have a solid foundation of Java SE even if you want to develop Java EE or Java ME applications

- Each Platform defines its own specifications (one or more).
  Specifications is just a document that describes the technology in plain english. And the software which is implementation of specifications is different & comes from different companies or providers (Oracle, Amazon ...)

- Java SE defines few specifications

Java Language Specification (JLS)
Java Virtual Machine Specification
Java API Specification

Different implementations of these specifications

- Oracle JDK
- Oracle OpenJDK (open source)
- Amazon Corretto
- Red Hat's open JDK

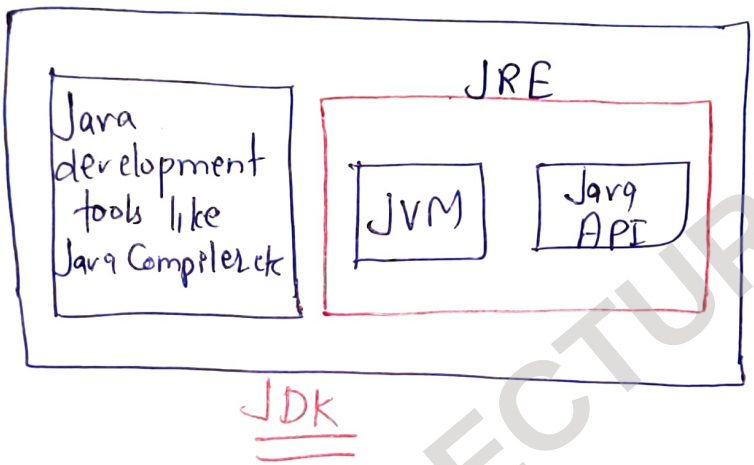
JLS:- defines the syntax, semantics & rules of Java Programming language. It is the most accurate document on Java as it is written by the Java language designers themselves like James Gosling.

- It contains detailed information on how Java behaves & how its various constructs should be implemented.

bcz: VMS :- defines how JVM should work & also specifies the bytecode instruction set.

Java API Specification :- specification of Java Library

→ There are many implementations of specifications and these implementation is basically JDK (Java Development kit) & we need JDK to write, Compile & execute our Java Programs.



So JDK is a software development kit/environment used for developing Java applications. It provides the necessary tools & lib. to write, Compile, debug & run Java programs.

It has :- Java Compiler (javac), JRE, JVM, Development tool (Java debugger, Java Doc, jar), and standard libraries. (Java API)

JRE :- It is a software package that provides the essential components needed to run Java applications. It includes JVM, core libraries & other resources necessary for executing Java Programs.

It can be installed separately/independently on a machine where you want to run Java applications or it can be included as a part of JDK installation.

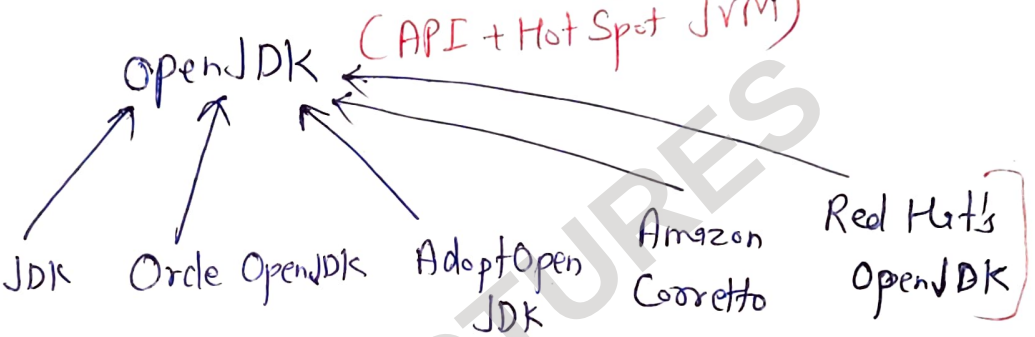
We can not develop Java applications in JRE.

So we want to ship our java applications to our clients or friends

then they would only need to install a JRE to run it.

→ Oracle used to offer such a JRE separately from JDK although JDK includes a JRE. Separate JRE was offered only till Java SE 10, from Java SE 11 onwards oracle stopped offering it as they were including a tool called jlink which can be used to create a custom JRE

→ But developers need JDK to develop java programs



↳ these are some prominent JDK providers.

Although these are different implementations, they are all part of same source code & this source code is developed as part of this project OpenJDK

OpenJDK is the official open-source version of the JDK. Anyone can contribute to it.

NOTE:- Java is Open Source

→ Let's see some Java SE Specifications.
↳ Google search Java SE Specifications

Now search Java 23 API / or Java 22 API or search for any Java version

⇒ Perform it on your Laptop

JSR:- Java Specification Requests: It describes the features that got added within that particular release

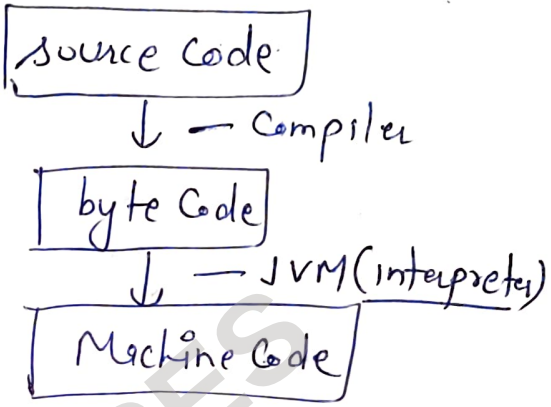
JCP:- Java Community Process :. A formal process to develop Java Specifications.

JDK

VM:- Java Virtual Machine

↳ It is responsible for running Java programs by converting Java bytecode into machine code that can be executed by the computer's processor.

JVM is not platform-independent but the Java bytecode it run is.

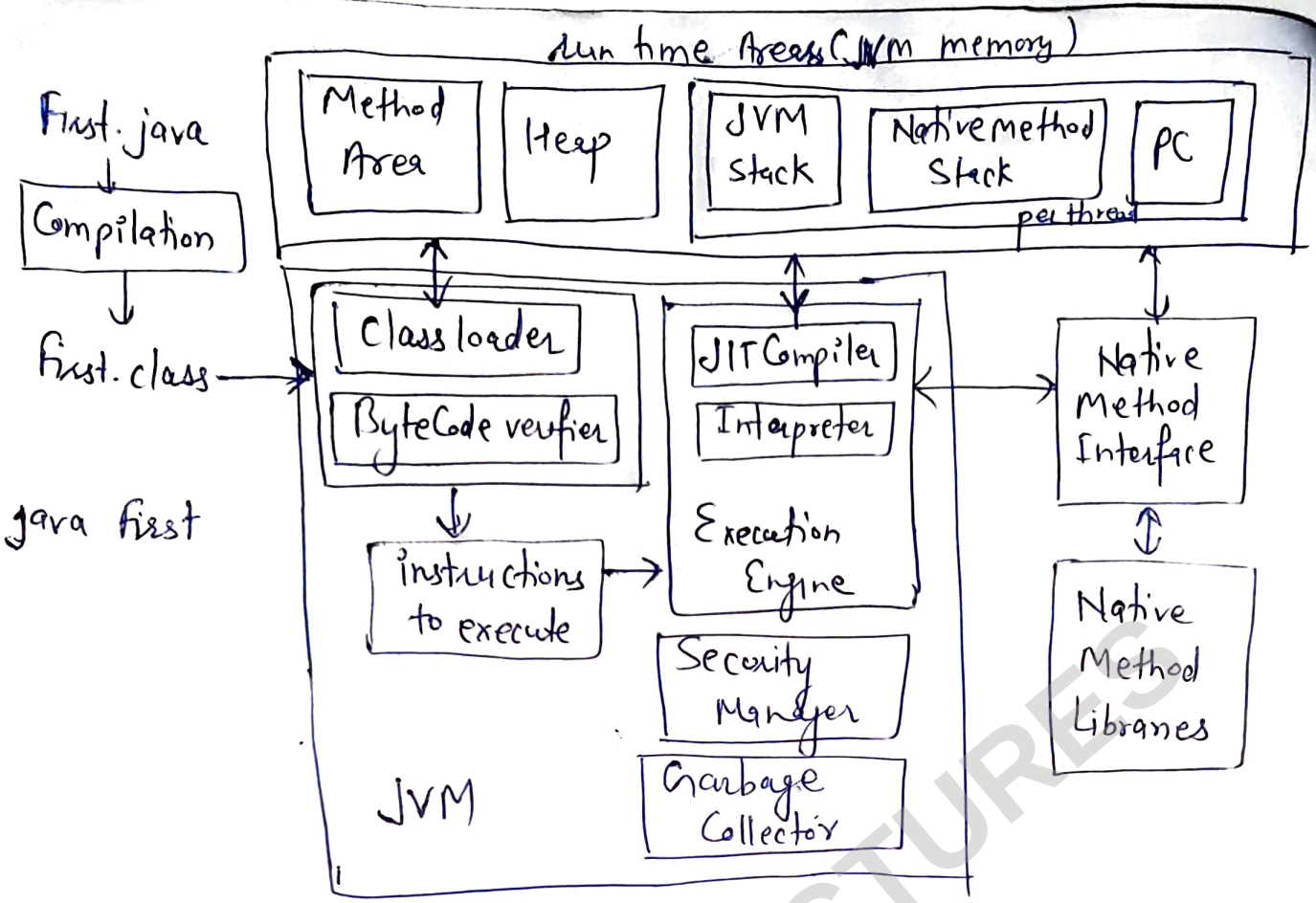


→ Different OS (windows, MacOS, Linux) & different processor architectures requires a different JVM implementation to interpret & execute Java bytecode.

→ Main Components of JVM:-

1. Class loader
2. bytecode verifier
3. Garbage Collector
4. Security Manager
5. Execution Engine (JIT + Interpreter)

JIT (Just-In-Time Compiler) is a component of JVM that improves the performance of Java applications by compiling Java bytecode into native machine code at runtime, just before execution.



HTML + Java Applets = Dynamic

Working :-

HTML page includes HTML tags corresponding to applets and the applets resides on some remote servers.

When web browser process HTML page & when it encounter with those HTML tag related to applet, it would download the corresponding applets from the remote server and then execute them as it also has Java installed.

Fast Execution (Java is fast)

① Bytecode optimization

↳ Java bytecode is compiled, compact & optimized for quick interpretion

② JIT Compilation :-

- Identify frequently executed bytecode known as "hotspots"

Also called as dynamic compilation

- Hotspots are given to JIT compiler & JIT compiler converts these hotspots into machine code & this machine code is cached (saved in memory for future use)

So in future if there is any need to execute these hotspots, the corresponding cached machine code will be executed & rest of the code is still interpreted by the interpreter.

Hotspot code will not be interpreted every single time