Introduction to Java.

A Brief History Story behind Java's Creation

* Sun microsystem, was a well-known American company that sold computers, computer components, software and Information Technology services 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 to brainstorm on this idea, so this Team decided to do something for electronic devices, home appliances (TV, setupbox, etc). So they named this project as "Green project"

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

* But there were some challenges in achieving this goal.

→ * As the devices (TV, VCR) have less/limited memory so the software should consume less memory

→ * These devices has their own H/w/platform. So the S/w program should be platform independent. So that it can run on any device.

* → Security (There S/W should not cause harm to the target devices)
* 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 Sheriden and Patrik Naughton, decided to create a new technology/language
* one of the teame member; James Gosling created an entirely new language and named it as Oak, after the tree outside his office.
* Games Gosling is the 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 S/W that could run on different computers connected via web (www)

their focus shifted from TV devices to web applications

* In 1995 - Oak was renamed as Java due to some trademark issues.

Why they name it Java?
Because of Java coffee which was popular among the team members during that time of development.
[Java is a 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 officially 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 prgm language. And this browser was designed to run Java applets (small, interactive progms that could be embedded in web pages to make them dynamic and interactive)

HTML + Java applet = dynamic

NOTE:- Applets have been removed from modern versions of Java because of security issues, declining usage, discontinued browser support. As now a days HTML 5, CSS, JS, PS being used to make interactive web pages.

But HotJava never became popular browser. Eventually other successful browsers like Netscape Navigator and Internet explorer adopted support for Java applets.
As the Internet expanded. so did Java's adaption by 1996. The first official release of Java (JDK 1.0) was out and it quickly became go to language during web applications. enterprise soln, desktop application & mobile applications.

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

Latest version of Java = Java 23.

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

→ Java Technology is both a programming language and a platform. Java PL is a high level object-oriented language that has a particular syntax and style.

* A Java platform is a particular environment in which Java pgming language applications run.
  * Java platform, Standard edition
  * Java platform, Enterprise edition
  * Java platform, micro edition.
  * Java fx.

Java SE is used to developing general purpose applications. It defines everything from basic types and object 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 (OS) H/w without modifications provided these is a compatible Java virtual machine available for that platform

Java follows WORA (write once Run Anywhere) principle means once you write and compile a Java prgm to bytecode. It can run on any platform using JVM.
This makes Java highly portable across OS.

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

3. Security :- Java applications can be downloaded from anywhere but it make use it does harm to your computer as the bytecode of the application run on ~~Java~~ JVM (Java virtual machine) JVM provides an isolated & restricted environment for running prgms.
* Security manager is a key feature in Java that defines access control rules for Java applications running in a restricted environment often refered to as a Sandbox.
* A Sandbox is a secure & isolated environment where code can be executed. Safely without

affecting the host ~~server~~ system
(or) other applications.

Eg:- modern web browsers use Sandboxing to run web applications and scripts.

mobile applications run in their own sandbox one app can't interface with another.

Dockes allow developers to create Sandboxed environments for applications. ensuring that dependencies & configurations don't affect other projects.

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

(4) 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 and pointers errors. means no need to add memory managemt logic.

ii. Strong Type checking :- Java 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 prgm can continue running smoothly instead of crashing.
Java has a strong exception handling and because of this the prgm can handle errors without crashing, providing meaningful feedback to the user.

Eg :- ClassNotFoundException, IoException etc.

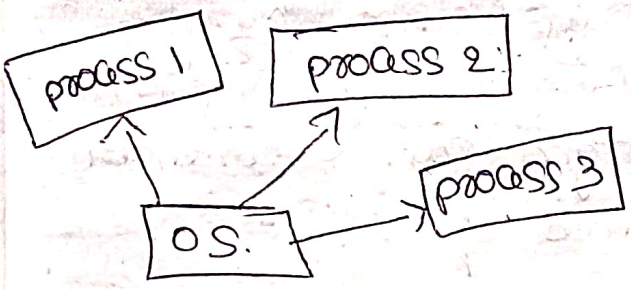
5. object - oriented programming :- Java supports oop features (Abstraction, encapsulation, Inheri-ence & polymorphism).

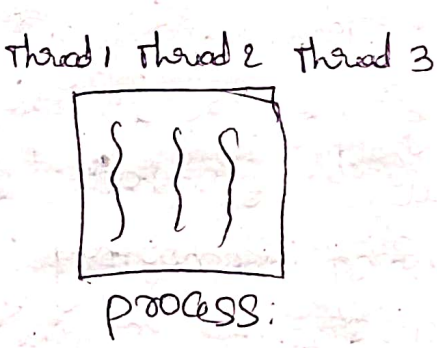
Java's object orientedness helps model real-world scenarios in a more natural way

6. Architectural neutral :- Java Compile generate bytecode that is architecture neutral, meaning it can run on any machine that has the JVM installed, regardless of the underlying H/w

7. multithreaded :- Java has built-in support for multithreading, allowing Concurrent execution of two or more parts of a prgm for maximum CPU utilization (threads)

In a restorant multiple waiters are there. Customers (task) 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 forground as an editor & another in the background auto saving the document, one thread to check the errors while you writing.

So thread is a unit of a process.

* multithreading used in webservers 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.

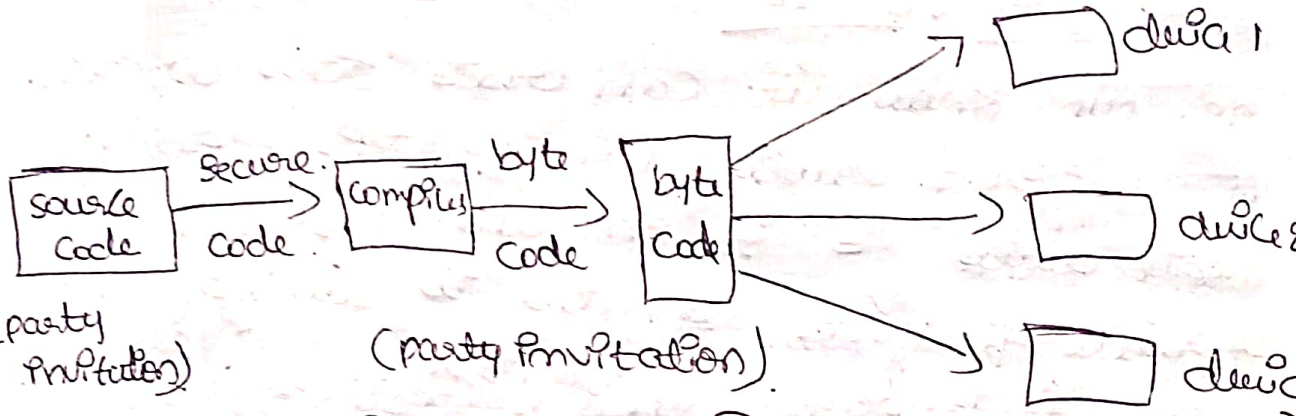
It is a general purpose, object-oriented high level language which is platform-Independed and very fast

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

How Java works

(Compilation)

Eg:- The goal is to make/write one application and have it work on whatever device your friend have.



① Java prgm. [.java file]

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

③ This compiled byte code is platform independent

④ All three devices should have JVM. to run this byte code

NOTE:- one more imp feature of Java is Backw-rd Compatibility. That means newer versions of Java language & JVM can run applications written in older versions of Java without repairing modifications. It is one of the reas 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, Mass. 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 desktop, webservers, mobile devices & so on. and those are separate platform dedicated to develop different type of applications.

Java SE:

editions / platforms of Java S/w family. mainly there are 3.

① Java Standard edition (previously called as J2SE) :- used to develop standalone application that typically run on desktops & servers.

Eg :- Hospital management System, (or) Hotel management System.

② Java Enterprise edition (previous called as J2EE) :- used to develop large scale applications runs on servers

Eg :- e-commerce website.

large no of users are accessing it at any particular time.

JSP & Servlets.

Java EE built on Java SE

Java EE has been rebranded as Jakarta EE because it has been shifted to Eclipse foundation

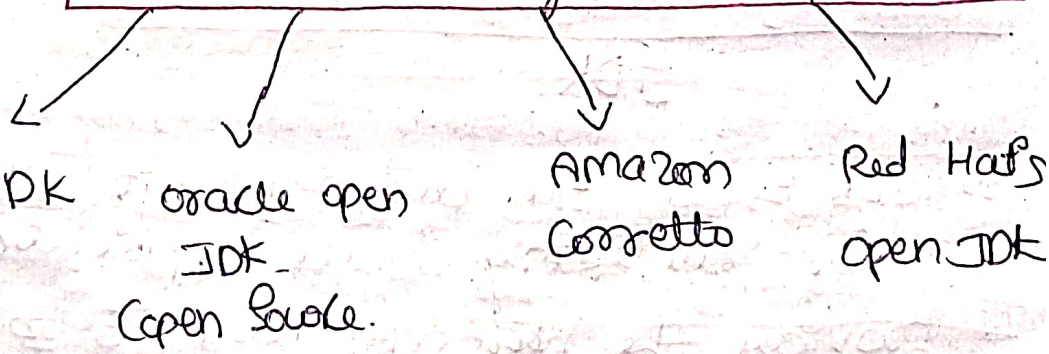
③ Java Micro Edition: (previously known as J2ME)
used to develop application for mobile, set up boxes (resource constraining device)
-> subset of Java ME

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 S/W which is implementation of specifications is different & comes from different companies (or) provides (oracle, Amazon..)

* Java SE defines few specifications.

Java language specification (aka JLS)
Java virtual machine specifications
Java API specification,



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

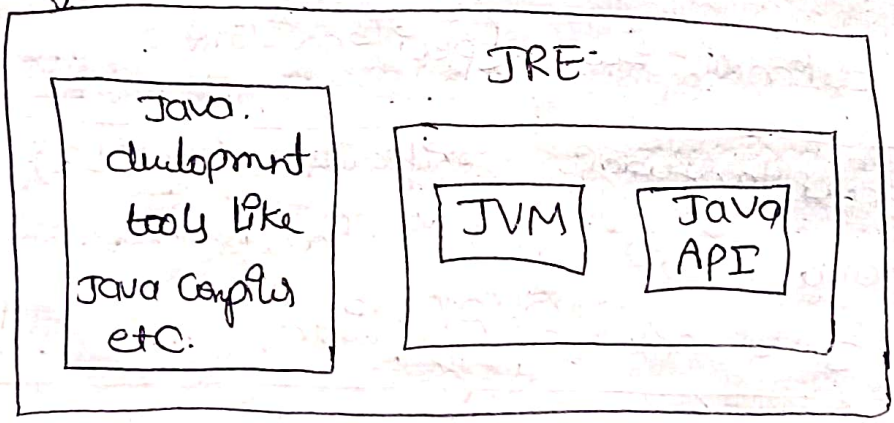
like James Gosling.

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

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 pgm.



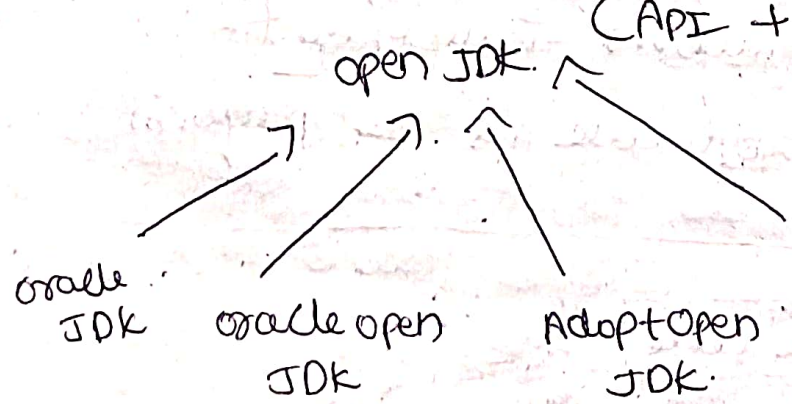
So JDK is a sw development kit/environment used for developing Java application. It provides the necessary tools & libs to write, compile, debug & run Java pgms. It has :- Java Compiler (javac), JRE, JVM, Development tools (Java debugger, Java doc, Jar) and Standard libraries (Java API)

JRE :- It is a sw package that provides the essential compiler needed to run Java applications. It includes JVM core libraries & other resources neccessary for executing Java prgms. It can be installed separately independently on a machine where you want to run Java applications. 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 seperately from JDK although JDK includes a JRE, separate 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 needs JDK to develop Java prgms.



There are some prominent JDK providers. Although there are different implementations, they are all part of same source code & this source code is developed as part of this project

openJDK is the official open-source version of the JDK anyone can contribute to it.

NOTE:- Java is a open source.

Lets see some Java SE specifications.

Google Search Java SE specification.

Now search Java 23 API (or) Java 22 API (or) search.

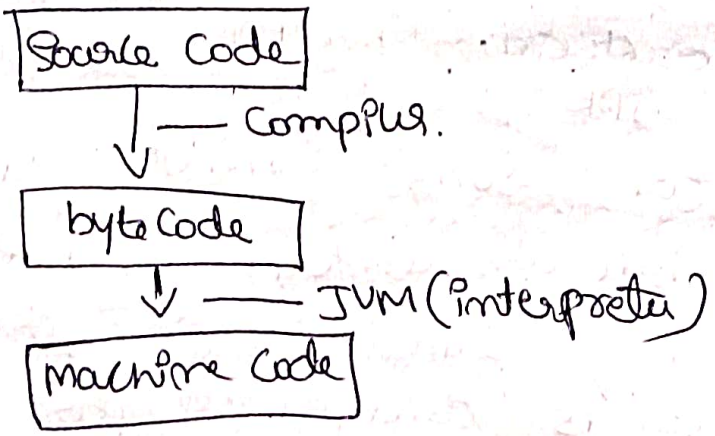
JSR → Java Specification Request: It depicts the features that get added written that particular release.

JCP :- Java Community process:- A formal process to adopt Java specifications

JVM :- Java virtual machine.

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

JVM is not a platform-independent-
But the Java bytecode is platform independent



→ Different OS (Windows, Mac OS, Linux) & different processor architectures requires different processor architectures & 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.