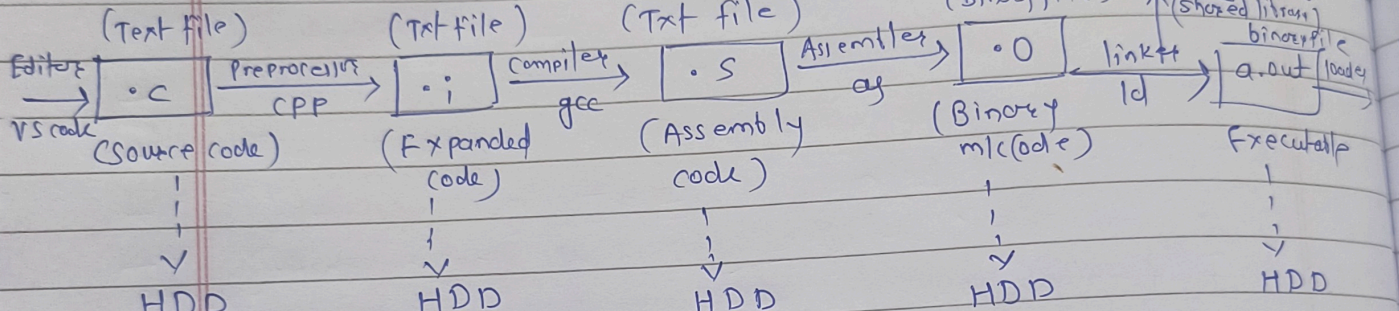
3/9/24 OOPJ

* Flow of Execution in C/C++

```
// File Name : main.c
#include <stdio.h>
int main (void) {
Printf ("Hello, World !\n");
 return 0;
}
```

Preprocessed file
gcc -E main.c -O main.i
// main.i is txt file
Assembly file
gcc -S main.c -O main.s
// main.s is a txt file
Executable
gcc main.c -O main.out
// main.out is a binary file



* Machine language code
hexadecimal code which understand by m/c (window OS
, mac, linux)

* Pros
Direct Execution
Performance

* Cons
complexity, portability, maintance, Development time

* Assembly language code
that is intermediate code (byte code)

* Pros
close to h/w, Performance

* Cons
it is same as above

* High level language code
It is Simple English language. (Syntax of lang)

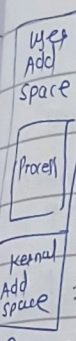
* Pros
Ease of use, faster development, portability. Error Handling

* Cons
Performance, memory usage, learning curve

Q How to make Programming language?
Syntax, Semantics, Data Types, Operator Set, Built-in Features.

- Use Case:
  CUI - Console User Interface Application.
  GUI - Graphical User Interface Application.
  Shared libraries (.jar)

We can create CUI, GUI & libraries using language but generally it is used to implement business logic
Ex - C, C++, Java, C#, Python, Go, Ruby etc



* Framework -
Set of ready made libraries on the top of it we can develop appln.

Q Why Framework?
→ Speed up development, Reduce errors, solve Specific prob
Ex- logging framework - Apache
Unit Testing - Junit
MVC based web appln framework - Apache Steuts
Automatic persistance framework - Hibernate

* Technology -
In general technology help us to develop appln / s/w
- Every lang can be considered as tehology but every technology can not be considered as language
Servlet / JSP, ASP.net etc
Developement, Deployment tools, Framework, Techniques

HLL - C english
LLL - assembly (main.exe) not readable by txt editor
MLL - binary/byte

Source file - portable
executable file - non portable

* platform -
A platform is h/w or s/w environment in which prog runs
Most plm can described as combinat° OS + H/w
h/w Based plm.
plm that do not required sp hardware but are built on top of existing OS & h/w s/w only plm
ex Java, Microsoft. Net, (jre)
In general s/w-only platform provides tools, API, Runtime enviroment.

* Introduction -
- Java is a Product of SUN/Oracle.
- Java language is both technology as well as platform.
- Java's Standardizatn is managed through the (Java Community Process JCP)
- Extension of Java source file ".java"
- Java is object oriented but also support Procedural & functional Programming paradigms.
- Java is statically typed language. It means type cheaking is done at compile time
- Java is case sensitive language
- Java, kotlin, Scala, Groovy are some of the JVM based language.
- ~~Java is platf~~

* Java Editions / Development platforms -
1. Java standard Edition (Java SE) - core Java
Java Appln programming Interface (API) & JVM
Used for standalone appln (desktop appln, cml tool)
2. Java Enterprise Edition (Java EE) - It called advance Java
web Java / Java, EJP, Java mail
JavaEE / JEE / Jakarta EE) • servelt, JSP, JPA, filter, JTA
Used for web appln and distributed appln.
3. Java Micro Edition (Java ME)
It is used to develop appln for mob phones, embedded system and IOT devices.

JDK for develop JRE for client
To compile Java cmd - Javac first.java/execute
To run Java - Java first

4. JavaFX
It is used for building rich, modern desktop appln UI for

5. Java Card
It is used for developing appln for smart card & secure IOT Devices

- API collects and processes a response, then returns with that response.

ex waiter, In hotel for dinner

* SDK (S/w Development kit)
To develop S/w developer must install the SDK on their m/c.
SDK = Dev Tools + API Docs + Supporting libraries + Execut^n Environment.

1) Dev Tools - Tools to compile, build, test & debug appln
2) Documentat^n - References that explain how to use SDK
3) Libraries - pre-written code which minimize developer efforts.
4) Execut^n Environment - plm where we can deploy & test the appln
5) Code Samples - ex projects that demonstrate how to implement certain features using the SDK.

* JDK (Java Development Kit)
To develop Java S/w, a developer must install the JDK on their m/c.

What is JDK?
→ Java SDK = Java Dev Tools + Java API Docs + Java Supporting lib. + Java execution environment
JDK = Java Dev Tools + Java API Docs + rt.jar + Java virtual machine
JDK = Java Dev Tools + Java API Doc + JRE (rt.jar + JVM)

In
- .class file inside compile / byte code present
- .class is binary file not read by txt editor
use Java p cmd

Java Development kit typically include
Java Dev Tools - javac, java, javap, jstack, jdb etc.
Java API Doc -
Java API Libraries - rt.jar
Execution Environment - JVM
Code Samples - src.zip

* JRE (Java Runtime Environment)
- To execute / run Java appl^n on developer's m/c / client's m/c, we require (JRE).
- The JRE comes with the JDK by default. So developers do not need to download it seperately.
- On client m/c we must first download & install the JRE
- Components of JRE
Java class Library (rt.jar) - to see native code
Java Virtual machine (JVM)

* JVM (Java Virtual Machine) -
- It is under the JRE.
It convert the compile byte code into Native code.
It work as interpreter between pl & underlying h/w
~~It consists of JIT compiler~~ It Provide jre to Java appln
~~To see native code use JIT watch~~ to run on different OS. / JIT compiler locate inside the JVM.
Jvm is foundat^n of Java PL.

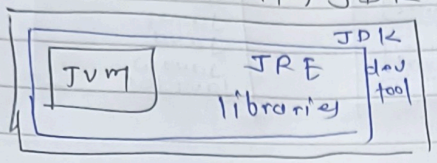
* JIT (compiler) present in JVM. (Just In Time)
It compile bytecode into m/c code at run time
It improve performance of Java program.
JIT compiler enable by default.
ex When method has been compile JVM calls the compile method instead of interpreting it

Stable JDK - 8

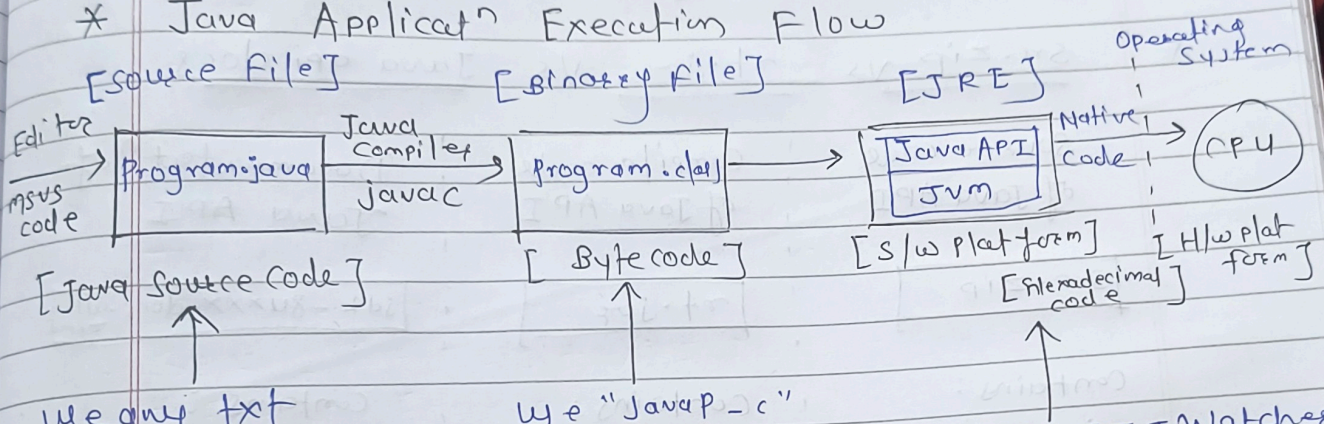
Mar - Sept new version release.

Now day JDK 23 come in industry

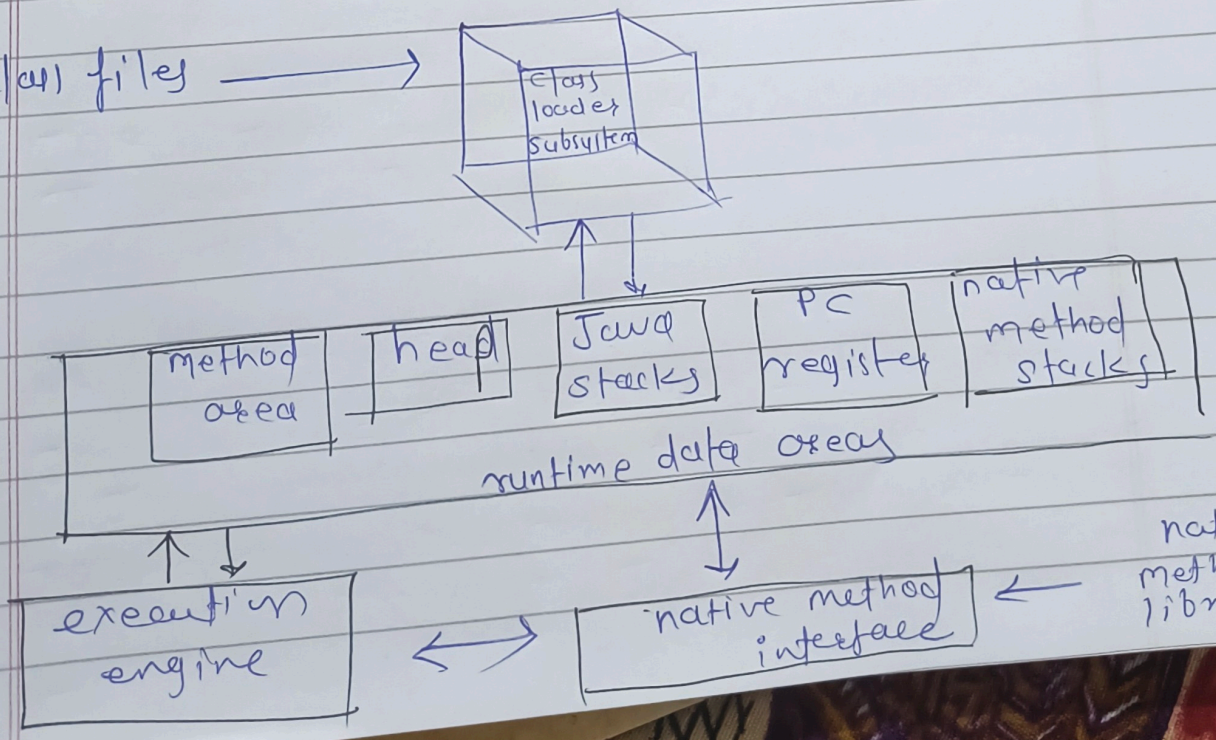
LTS (long term support) we in IT industry are JDK 8, JDK 11, JDK 17, JDK 21

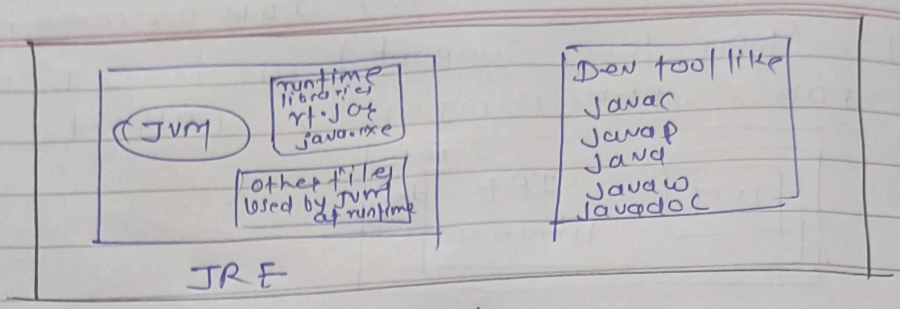


* Java Applicatn Execution Flow



* Overview of JVM Architecture.





* src.zip vs rt.jar Vs Java api docs

Source code of Java API	Compile code of Java API	Documentation of Java API
src.zip	rt.jar	jdk-8uxxx-docs-all.zip
Contains: java files	Contains: .class file	Contains: .html files