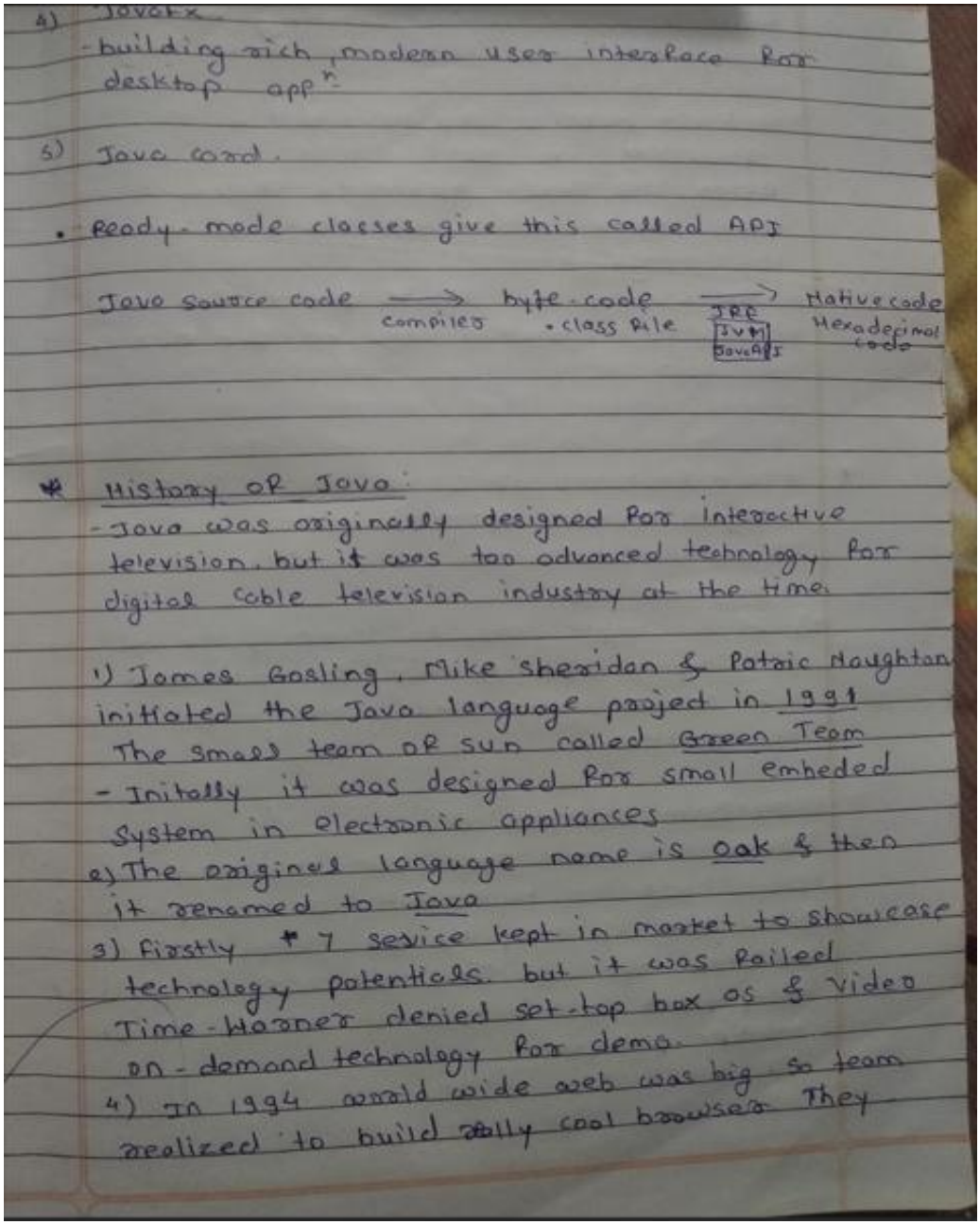
1. Reading Assignment: A Short History of Java

- **Task**: Read about the history and development of Java.
- **Link**: http://sunsite.uakom.sk/sunworldonline/swol-07-1995/swol-07-java.html



4) JavaFx
- building rich, modern user interface for desktop app^n

5) Java card.

- Ready-made classes give this called API

Java source code → (compiled) byte-code .class file → (JRE: JVM, Java API) Native code Hexadecimal code

* History of Java:
- Java was originally designed for interactive television, but it was too advanced technology for digital cable television industry at the time.

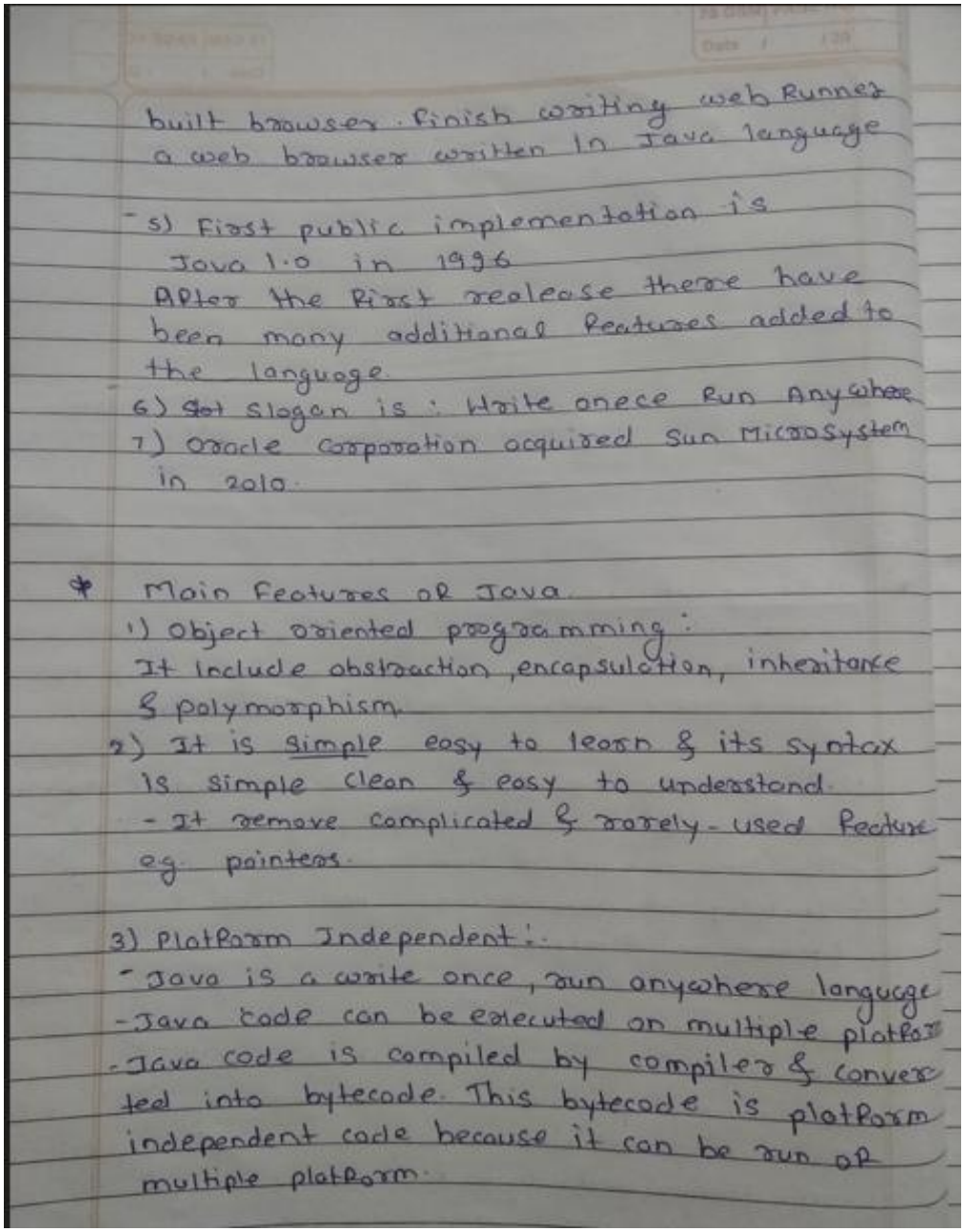
1) James Gosling, Mike Sheridan & Patric Naughton initiated the Java language project in 1991
The small team of sun called Green Team
- Initially it was designed for small embeded system in electronic appliances
2) The original language name is oak & then it renamed to Java
3) Firstly * 7 sevice kept in market to showcase technology potentials. but it was Failed Time-Warner denied set-top box os & video on-demand technology for demo.
4) In 1994 world wide web was big so team realized to build fully cool browsers They

built browser. Finish writing web Runner a web browser written in Java language

5) First public implementation is Java 1.0 in 1996
After the first realease there have been many additional features added to the language.
6) ~~Got~~ Slogan is : Write onece Run Anywhere
7) Oracle corporation acquired Sun MicroSystem in 2010.

2. Reading Assignment: Java Language Features

- **Task**: Learn about the main features of Java.
- **Link**: https://javaalmanac.io/features/



built browser. Finish writing web Runner a web browser written in Java language

5) First public implementation is Java 1.0 in 1996
After the first release there have been many additional features added to the language.
6) Slogan is : Write once Run Anywhere
7) Oracle Corporation acquired Sun MicroSystem in 2010.

* Main Features of Java.
1) Object oriented programming:
It include abstraction, encapsulation, inheritance & polymorphism.
2) It is simple easy to learn & its syntax is simple clean & easy to understand.
- It remove complicated & rarely-used feature eg. pointers.

3) PlatForm Independent:
- Java is a write once, run anywhere language
- Java code can be executed on multiple platfor
- Java code is compiled by compiler & conver ted into bytecode. This bytecode is platform independent code because it can be run on multiple platform.

4) Secured :
Java is best known for its security. It is secured because
- No explicit pointer
- Java program run inside a virtual machine sandbox

5) Robust (strong)
- It uses strong memory management.
- There is a lack of pointers that avoid security problems.
- It provide automatic garbage collection which run on Jvm

6) Portable :
- Java is portable because it facilitates you to carry the Java bytecode to any platform.

7) High performance.

3. Reading Assignment: Which Version of JDK Should I Use?

- **Task**: Find out which JDK version is right for you.
- **Link**: https://whichjdk.com/
-
- JDK release model, a new release with a new major version number is planned with every six month in march and september.

JDK Version	Type	Release Date	Highlights
8	**LTS**	**03/2014**	Lambdas
9	Feature	09/2017	Modules
10	Feature	03/2018	var
11	**LTS**	**09/2018**	New HTTP Client
12	Feature	03/2019	
13	Feature	09/2019	
14	Feature	03/2020	Switch expressions
15	Feature	09/2020	Text blocks
16	Feature	03/2021	Records
17	**LTS**	**09/2021**	Sealed Classes
18	Feature	03/2022	UTF-8 by Default
19	Feature	09/2022	
20	Feature	03/2023	
21	**LTS**	**09/2023**	Pattern Matching, Virtual Threads
22	Feature	03/2024	_

LTS version is useful it give long term support Current LTS version is 21.

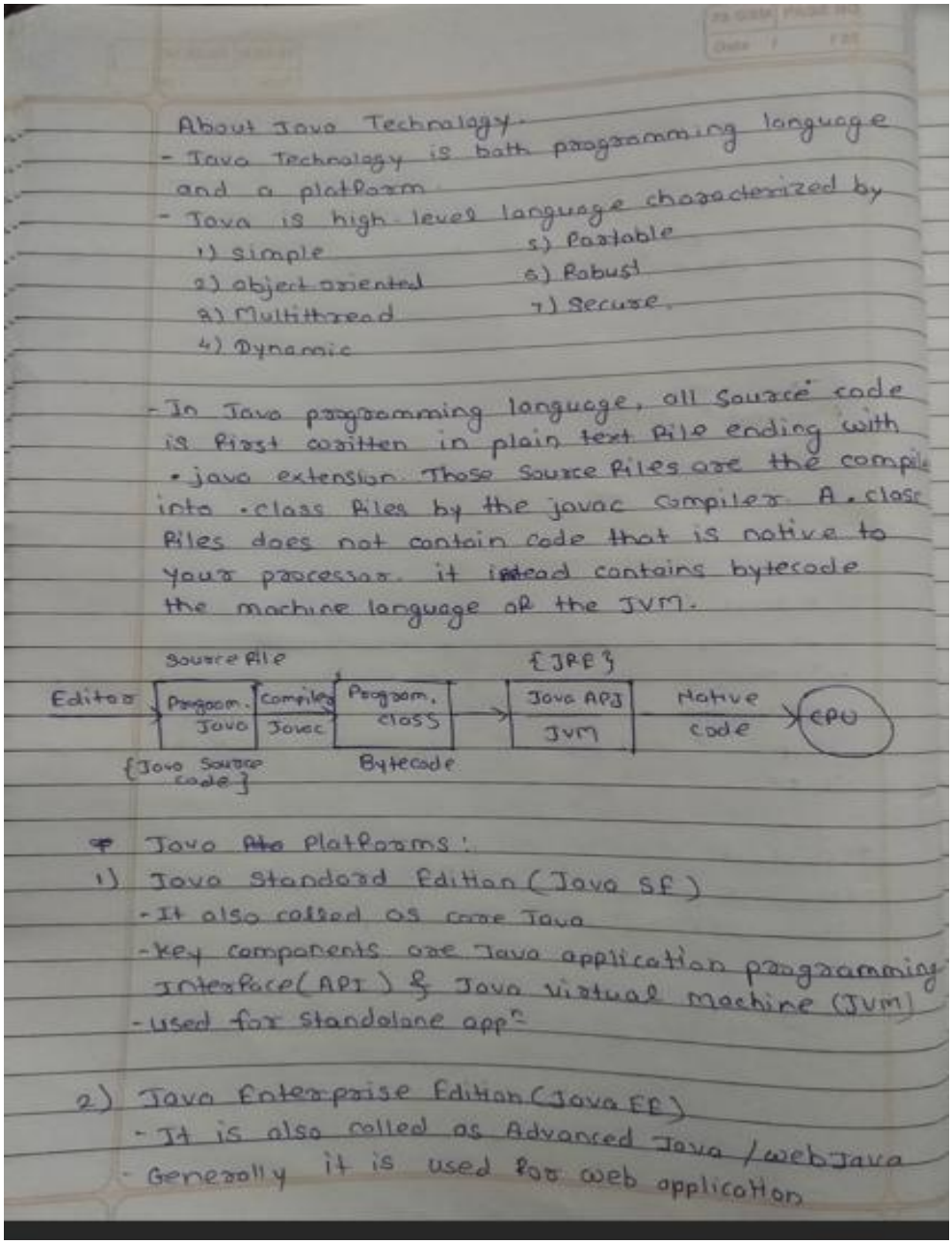
4. Reading Assignment: JDK Installation Directory Structure

- **Task**: Understand the folder structure and files in the JDK installation.
- **Link**: https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jdkfiles.html

jdk-1.8
- bin
- libr
- jre
  - bin
  - lib
    - ext
    - fonts
    - security
    - sparc
    - rt.jar
    - charsets.jar

5. Reading Assignment: About Java Technology

- **Task**: Read about the basics of Java technology and its components.
- **Link**: https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html



About Java Technology.

- Java Technology is both programming language and a platform
- Java is high level language characterized by
  1) Simple
  2) object oriented
  3) Multithread
  4) Dynamic
  5) Portable
  6) Robust
  7) Secure.

- In Java programming language, all source code is first written in plain text file ending with .java extension. Those source files are the compile into .class files by the javac compiler. A .class files does not contain code that is native to your processor. it instead contains bytecode the machine language of the JVM.

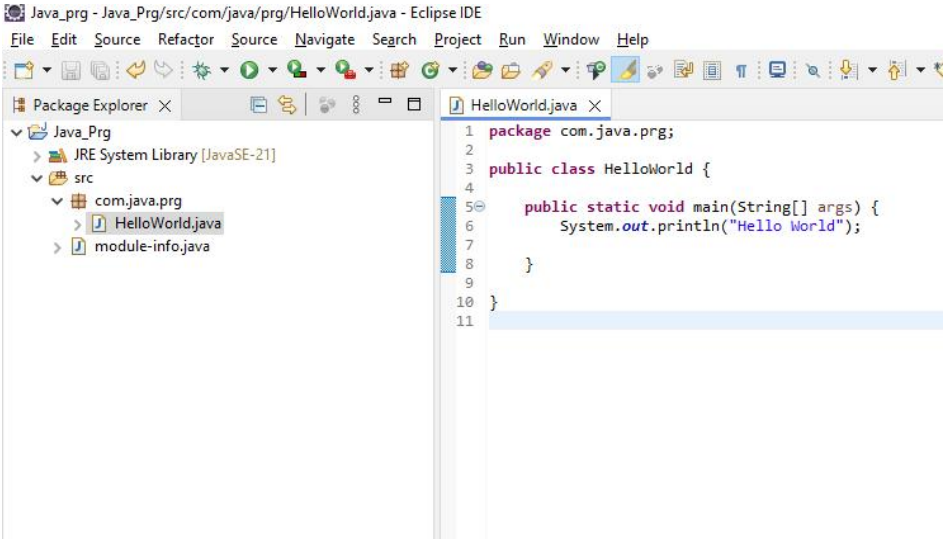
Java Platforms:

1) Java Standard Edition (Java SE)
   - It also called as core Java
   - key components are Java application programming interface (API) & Java virtual machine (JVM)
   - used for standalone app^n

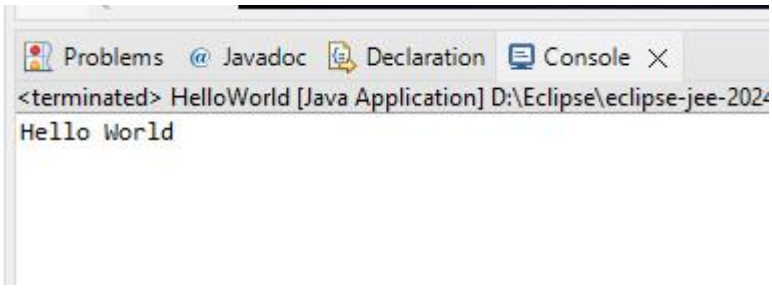
2) Java Enterprise Edition (Java EE)
   - It is also called as Advanced Java / web Java
   - Generally it is used for web application

6. Coding Assignments

1. **Hello World Program**: Write a Java program that prints "Hello World!!" to the console.

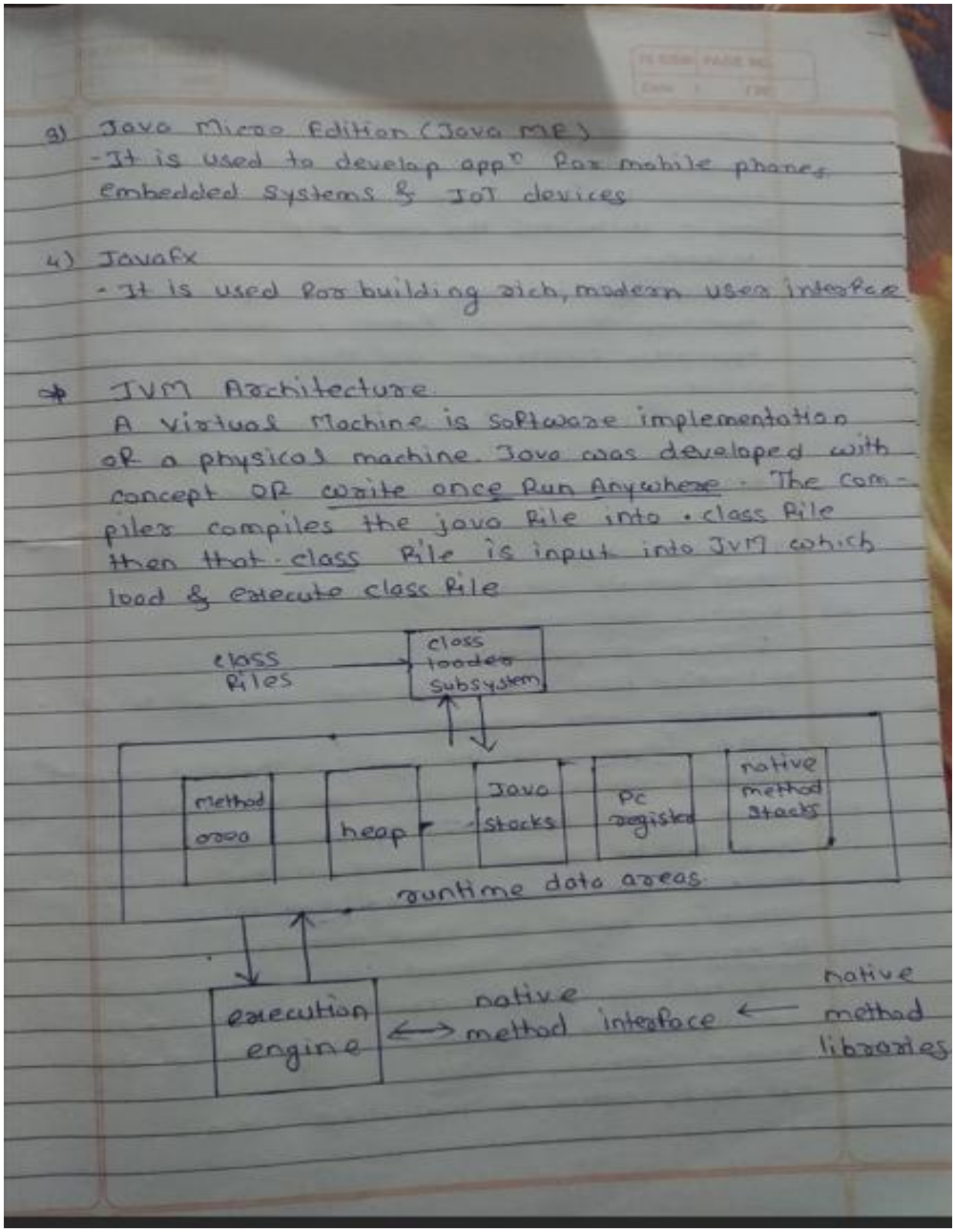


Output



7. Reading Assignment: The JVM Architecture Explained

- **Task**: Learn about how the Java Virtual Machine (JVM) works.
- **Link**: https://dzone.com/articles/jvm-architecture-explained



1) classloader Subsystem
- Java's dynamic class loading functionality is handled by the classloader subsystem. It loads, links & initializes the class file

1) Loading
classes will be loaded by this component.
Bootstrap classloader, Extension classloader, Application classloader are three will help achieve it

2) Linking
- Verify : it verify bytecode is proper or not
- prepare : all static variable assign default value
- Resolve : Symbolic memory ref. replaced with original ref

3) Initialization :
- Static variable assign original values

2) Runtime Data area :
5 main components
1) Method Area : All class level data will be stored including static variables.
2) Heap Area : All the objects & their corresponding instance variables & arrays will be stored
3) Stack Area : For every thread, Separate runtime stack will be created for every method call one entry made in Stack memory called Stack Frame. Local variable created in stack memory.
4) PC Registers : each thread separate PC Register to hold address of current executing instruction once instruction executed PC register updated.
5) Native Methods Stacks
- It holds native method information.

3) Execute Engine
- The bytecode, which is assigned to the runtime data area, will be executed by Execution Engine
- It reads byte code & execute it piece by piece.

- Java Native interface (JNI) - It will interacting with Native method libraries & provide the native libraries for execution engine
- Native Method libraries : collection of libraries which is required for execution engine