

①

Explain the component of the JDK

→ JDK is a software development kit that provides the necessary tools and libraries to develop Java application. It includes several components essential for Java development.

①

Java compiler (Java C)

②

Java Runtime Environment (JRE)

③

JVM (Java Virtual machine)

④

Java API libraries

⑤

Development Tools :- JDK includes various development tools to aid in the software development process.

①

Java Debugger (jdb) :- Allows developer to debug Java application interactively.

②

Java Doc :- Used to generate API documentation in HTML format from Java source code comments.

③

Java Archive (JAR) :- A utility for packaging Java classes and resources into a single compressed file, facilitating distribution and deployment of Java application.

(4) JavaFX compiler and runtime :- by using JavaFX for building graphical user interfaces, the JDK provides tools and libraries for developing JavaFX applications.

(5) JavaFX :-

- 1) Starting from Java 11, JavaFX is no longer bundled with the JDK. However, newer versions of JDK include JavaFX, which is a platform for creating rich client applications using Java.
- 2) JavaFX provides a set of API for creating modern UIs, animation, multimedia and more.

(Q2)

Differentiate between JDK, JVm, JRE

(1) JVm :-

- ① It is an abstract computing machine that provides the runtime environment for executing Java bytecode.
- ② It interprets or compiles Java bytecode into native machine code for the underlying platform.
- ③ JVm manages memory, handles exceptions, performs other runtime tasks.
- ④ Multiple implementations of JVm exist, each optimized for different platforms.

## ② JRE (Java Runtime Environment)

- ① The JRE is a part of the JDK and a standalone installation option.
- ② It provides the runtime environment for executing java applications.
- ③ JRE includes the Java Virtual Machine (JVM) along with core libraries and other components required for running java application.
- ④ Users need JRE installed on their system to run java application, it does not include development tools like JDK.

③ What is the role of the JVM in Java?  
How does the JVM execute Java code?

- i) JVM plays a central role in the Java ecosystem.
- ii) Function is to provide a runtime environment for executing Java bytecode.

### \* Role of JVM:

- ① Platform Independence & One of the main features of Java.
- ② Memory management.
- ③ Execution of Bytecode.
- ④ Exception handling.
- ⑤ Security.

## \* How JVM executes Java code!

(1)

### Loading :-

When a java program is executed, the JVM loads the byte code of the program into memory from the file system or network.

(2)

### Verification:-

The JVM performs bytecode verification to ensure that the loaded bytecode adheres to the Java language specification and does not violate security constraints. This step helps prevent malicious code from causing harm.

(3)

Execution :- The JVM executes bytecode either by interpreting it or by using just-in-time (JIT) compilation technique to translate bytecode into native machine code per the host system.

(4)

### Memory management:-

The JVM manages memory allocation and garbage collection. It allocates memory for objects created during program execution and reclaims memory occupied by objects that are no longer reachable, thus ensuring efficient memory usage.

(s) Exception Handling :- During execution, the JVM detects and handles exception, ensuring that program can gracefully recover from errors and continue execution or terminate in a controlled manner.

(ii) Explain the memory management system of JVM

→ (i) Memory Areas:-

① Heap :- The heap is the primary memory area used by the JVM for storing object created by the Java application.

StringBuilder sb = new StringBuilder();

The above statement creates an object of the StringBuilder class. The object allocated to the heap, reference sb allocate to stack.

Heap is divided into:-

- ① young generation
- ② survivor space
- ③ old generation
- ④ permanent generation

② Stack :- threads in Java application has its own stack, which stores method invocations, local variables, intermediate result. It smaller in size compared to the heap and is not shared between threads.

③ program counter : PC ~~loop~~ register keep track of the current execution point for byte code of the currently executing thread.

② Garbage collection :-

① Garbage collection is the process by which the JVM automatically reclaims memory occupied by objects that are longer in use.

② JVM uses different garbage collection algorithms :-

① young generation Garbage collection.

② old generation Garbage collection.

③ Memory optimization techniques:-

① Object pooling :- Reusing object instead of creating new one can reduce memory allocation and deallocation overhead.

② Tuning Garbage collection :- Adjusting JVM settings can optimize memory usage and garbage collection performance.

③ Memory leak detection :- Profiling tool can help to identify memory leaks by analyzing memory usage patterns and object retention.

Q) what are the JIT compiler and its role in the JVM? What is the bytecode and why is it important for Java?

## → Q) JIT Compiler :-

① Role :- Just in Time compiler is a component of the JVM that converts Java bytecode into native machine code during runtime.

## ② Bytecode :- purpose :-

It enhances java application's performance by dynamically optimizing frequently executed bytecode segment.

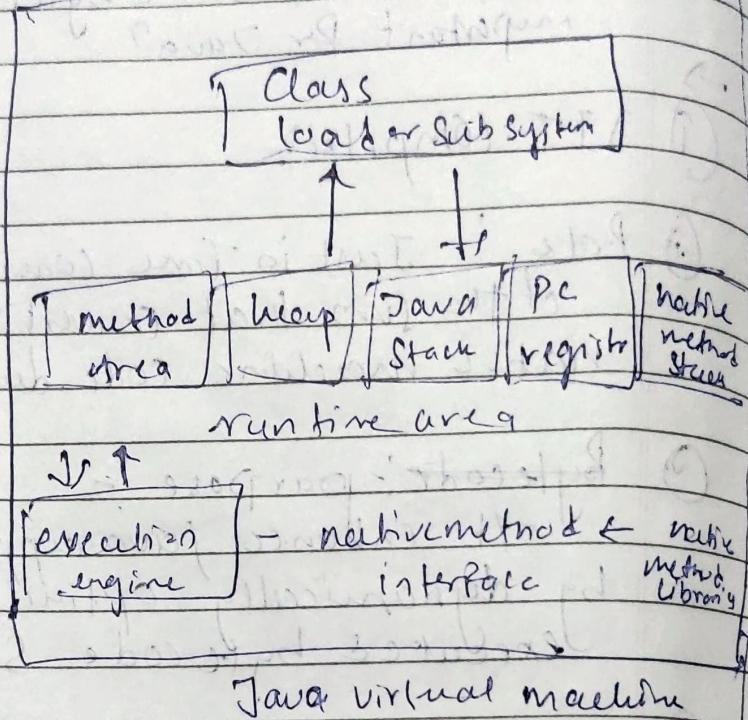
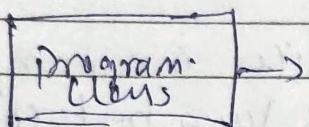
## ③ Bytecode :-

(i) Definition :- Bytecode is an intermediate representation of java source code compiled by the Java compiler.

④ Importance :- It enables platform independence in java, allowing bytecode to run on any device with compatible JVM, facilitating "write once, run anywhere" capability.

⑥ Describe the architecture of JVM:-

Bytecode



Java virtual machine

\* Components of java virtual machine :-

① Class loader sub system

- 1) Bootstrap class loader
- 2) Extension class loader
- 3) System class loader
- 4) Custom class loader

② Runtime data area

- 1) method Area
- 2) Heap
- 3) Java stacks
- 4) PC Register
- 5) Native method stack

- ⑥ Execution engine is:  
① interpreter  
② Just-in-Time (JIT) Compiler  
③ garbage collector.

⑦ How does Java achieve platform independence through the JVM?

→ By using Bytecode and the JVM, Java achieves platform independence, allowing Java applications to run unchanged on any device or operating system with a compatible JVM. This "write once, run anywhere".

⑧ What is the significance of the class loader in Java? What is the process of garbage collection in Java.

→ ① Significance of class loader:-

① Dynamic class loading:-

The class loader is responsible for dynamically loading these classes into memory as they are referenced during runtime.

② Classpath management:- class loader manages the class path, which is a list of directories and JAR files where Java classes are stored. They locate and load classes from the classpath.

③ Namespace Isolation : each class loader contains its own namespaces preventing naming conflicts between classes loaded by different class loader.

④ Customization : java provides mechanism for custom class loader, allowing developer to load classes from non-standard sources.

\* The garbage collector in Java periodically scans the heap memory to find objects that aren't being used. The process of garbage collection involves several steps including marking, sweeping, compacting.