

### Assignment-3

#### Interview Question

Components of JDK?

JDK (Java Development Kit) consists of many components:

- java Compiler (javac) - this tool converts Java Source code files (.java) into byte code (.class) file which can be understood by the JVM.

Java Runtime Environment (JRE) - the JRE includes the Java Virtual Machine (JVM) and the class libraries necessary for running Java application.

Development Tools - The JDK also provides various tools for Java development such as:

- JavaDoc - Generate API documentation from Java Src code
- Jar - Create Jar (Java Archive) files for packaging multiple Java class files & resources
- Debugger - A tool for finding & fixing Java errors

Difference b/w JVM, JDK and JRE?

JDK - It is comprehensive tool kit for debugging, compiling & running Java applications.

JVM - The execution engine that interprets and runs the Java byte code.

JRE - the minimum set of component required to run Java applications, including JVM and class libraries.

JDK includes the JRE and addition of tools for developers, while the JRE is the core component needed to run Java Program & JVM is starting point of running

2)

Role of the JVM ? crucial role in Java Platform

The JVM plays a execution  
independence

3)

Bytecode Interpretation - The JVM represents the bytecode instruction &  
which is the intermediate code which can be executed  
src code , if interpret the  
execute them .

4)

JIT Compilation - To improve the performance moden  
JVM use JIT compilation this involves  
compiling frequently executed bytecode section  
compiling native machine code which can be executed  
into native machine more efficiently

Memory Management - The JVM manage memory for Java  
appln . It allocates memory for objects on heap  
and automatically ~~free~~ unused memory through  
gc

Garbage Collection - The JVM collector identifies the object  
that are no longer use & free up memory  
they occupy this process helps prevent  
memory leakage & ensure efficient of memory

5) JVM Memory Management ?

The JVM uses set of memory areas

6)

Heap - The primary memory area for  
object allocation . Objects are created  
on the heap & their reference are stored  
on the stack .

### classmate



Stack — Used for storing method calls, local variables & return values each method call creates a new stack frame

Method Area — Stores class metadata, including the class bytecode, static variables & method information

Registers — Used to keep track of current instruction being executed and current memory management system allocates and deallocates memory for these areas as needed. GC is used to reclaim memory on the heap that is no longer in use.

JIT Compiler and Bytecode? →  
Bytecode — The intermediate language produced by the Java compiler. It is platform independent and can be executed by any JVM.  
JIT Compiler — A component of JVM that can compile frequently executed bytecode sections into native machine code. This improves performance by eliminating the overhead of interpretation.  
JIT compiler enhances the performance of Java applications especially for long running programs. By compiling frequently used code into native machine code JVM can avoid the overhead of interpretation & execute the code more efficiently.

JVM Architecture

JVM Architecture consists of key components

Q) For loading class file in Java file into memory It searches for class files that verify their integrity.

- 1) Class loader : Responsible for class file in Java file into memory
- 2) Execution Engine : The core component that may use JIT compilation to optimize performance of interpreted byte code
- 3) Runtime Data Area ; It includes PC register which heap, stack, method area, and program execute are used to store data during program execution
- 4) Garbage collector : Automatically reclaim memory that is no longer in use

Q) Platform independence ?

The JVM achieves P.I. independence by providing a consistent execution environment across different OS. Bytecode - The JVM interprets bytecode which is P.I.D representation of code.

Abstract machine - The JVM acts as abstract machine that hide the how & os detail from Java program.

Standard library - JVM includes std lib that provide a consistent set of API's for common tasks, regardless of underlying platform.

Q) Class loader and Garbage Collection ?

- 1) Class loader :- Loads class files into memory as needed . It uses a hierarchical structure to search for class in different directory or JAR file.
- 2) Garbage collection :- Automatically reclaim memory that is no longer in use . The JVM's GC collector uses algorithm like mark and sweep or generational collection to identify and reclaim unused objects.

classmate

classmate



The class loader & garbage collector work together to manage the JVM's memory & ensure that classes are loaded and unloaded efficiently.

TOPIC :-  
↳ Inheritance  
↳ Methods  
↳ Constructors  
↳ Overloading  
↳ Overriding  
↳ Polymorphism  
↳ Interfaces  
↳ Abstract Classes  
↳ Final Classes

- Four access modifiers in Java?
  - ↳ Java provides four access modifiers to control the visibility of classes, methods and variables.
- Public - Accessible from anywhere, including other pkgs and classes.
- Protected - Accessible within the same pkg or subclass.
- Default (pkg private) - Accessible only within the same class.

↳

- Difference b/w public, protected and default?
  - ↳ Public - The most permissive allowing access from anywhere.
  - Protected - Grant access to the same pkg and subclasses to default (pkg private) - Restricts access to subclasses to the same pkg.

↳

- Overriding with Different Access Specifiers
  - ↳ Yes you can override a method in a subclass of program with different access modifier, but only with a more permissive modifier ex. a Protected method in a Superclass can be overridden with a Public method in a subclass, but not vice versa.

↳

- Difference b/w Protected and Default ? It allows access to subclasse.
- Difference b/w Protected and Subclass.
  - ↳ Protected - grant access to the same pkg & subclass.
  - Default (pkg private) - restricts access to the same pkg, not allowing access to subclasse.

Q Private class ?  
→ If it is not possible to make a class private in Java. A class must be declared as public or default (pkg - private)

Q What level class access modifiers ?

- Top level class in Java can only be declared as top-level class ( pkg - private ) Protected by Public or default or not allowed for top level class

Q Accessing Private members from Another class ?  
→ If you declare a variable or a method as private in a class, you cannot directly access it from another class within the same pkg . The visibility is restricted to the declaring class .

Q Pkg - private Access ?

→ Pkg private ( or default ) access means that a class method or variable is accessible only within the same pkg . This provide a middle ground between public & private visibility , allowing for controlled access within a pkg but preventing access from outside .