- Interview questions.

① Components of JDK →

JDK → software environment used to develop java applications.

- Compiler (javac) →
  converts java src code into bytecode.
- JRE →
  Contains libraries, JVM and components neccessary to run java applications
- JVM → Execute java bytecode.
- jdb (javadeguer :
- javadoc
- JavaFX, java Mission Control & other tools.

② Differente JDK, JRE & JVM →

JDK	JRE	JVM
1) Java dev kit	Java Runtime envi.	Java Virtual M/c.
2) tool neccessary to compile, doc & package java pgms.	2) refers to a runtime enviro in which java bytecode can be executed.	2) abstract m/c. It is a specification that provides a run-time environment in which java bytecode can be executed.
3) Contains JRE + dev tools.	3) implement of ~~java~~ jvm which physically exists.	3) JVM follows three notations → specification, Implementation & Runtime instance

2) Role of the JVM in Java & How does the JVM execute java code?
→ Role → JVM allows java ~~allows~~ appli-cations to run on any platform without modification, making Java platform-independent.

Execution Process →
① Class loader → loads java bytecode into JVM.
② Bytecode Verifier →
checks correctness of bytecode to prevent security violations.

③ Execution Engine →
Converts bytecode into m/c code using an interpreter, ~~&~~ JIT compiler then executes it on the host m/c.

④ Memory management in the JVM →
→ • Heap → stores obj. & their instances
• Stack → Method store call frames, local vari & intermediate results
• Method area → store class structure
• pgm counter reg → track of address of current exe.
• ~~Network~~ Native method stack →
supports native (non-Java) metho exe.

⑤ JIT compiler & Bytecode →

→ • JIT compiler (Just-in-Time)
- part of JVM.
- improves performance by compiling bytecode into native m/c code at runtime.
- Various uses optimization techniques

⑥ Architecture of JVM →

• Bytecode → generated by Java compiler
platform independent.
runs on JVM.

→ 1) Class loader pgm → loads, links.
2) Runtime Data Areas → Memory areas where data is stored during exe (heap, stack)
3) Execution Engine →
Execute bytecodes using Interpreter.
4) Garbage collector →
Manages memory by automatically
5) Native interface →
Allow java to interact with native libraries.

⑦ platform Independence of Java through JVM.

→ compiling source code into bytecode which is platform agnostic. The JVM on each platform translates this bytecode into m/c specific instn^s. allowing same Java pgm

⑧ class loader & garbage collector.
→ a) class loader →
Loads classes into JVM during runtime. Handles three main process loading, linking & initializing classes. Types includes Bootstrap, extension & appⁿ class loaders.

b) Garbage collⁿ →
automatic mem mgt process that reclaims memory used by objects no longer referenced.

⑨ ~~Plas~~ four access modifier →
→1) Public → Accessible from any class,
2) protected → assessible in same package & by subclass.

3) Default →
Accessible only within the same Package.
4) Private → accessible only the same class.

classmate
Date
Page

(10) Diff betn public, Protected 4 default access modifier.
→ 1) Public →
Members are accessible from any class in any package.
2) Protected →
Within the same package 4 subclasses in other packages.

3) Default →
Only within classes in the same Package.

(11) Overriding Methods with different access modifiers.
→ You cannot override a method with a more restrictive access modifier in a subclass.

For instance, a protected method in superclass cannot be overridden with a private method in a subclass.
The overriden method must mainta the same or more permissive acces level.

(12) What is the diff. betn protected 4 default (package - private) access →
→ 1) protected →
Access within same package 4 by subclass even in other packages.

b) default (package - private) →
access only within the classes of
the same package.

Q) Is it possible to make a class private
in Java? if yes, where can it be
done and limitations?
→ A private class is allowed only as
an inner class within another class.
A top - level class cannot the private

A Private inner class is access only
in same class., limited scope.

Q) Can a top-level class in java be
declared a variable or method as
Private or protected? why or why
not?
→ Top-level classes in Java cannot be
declared as protected or private;
they can only be public or package-
private.

Q) What happens if you declare a
variable or method as private in
a class & try to access it from
another class within the same
package.
→ If you declare a variable or
method as private in a class, it
cannot be accessed from any other
class, even within the same

package.

(15) Explain the concept of "package-priv
ate" or "default" access. How does it
effect the visibility of class member

→ When no access modifier is speci
members have package-private
access, meaning they are accessible
only within classes in the same
Package.

The access level provides more
control over visibility of class mem
bers than public, protected
without exposing them outside
the package.