Q.1 Explain the components of the JDK.

→ # Java Compiler (javac):
- Converts Java source code into bytecode.

Java Runtime Environment (JRE):
Provides libraries & JVM to Run Java appln

Java debugger (jdb):
Used for debuging Java programs.

javaDoc:
Generate API documentation from Java source code.

Java Archive (Jar)
packages Java classes into a single JAR File.

Q.2 Differentiate bet^n JDK, JVM & JRE?

→

JDK	JVM	JRE
• Java developing kit.	• It Java virtual machine.	• Java Runtime Environment.
• It includes everything needing to write, compile, & debug For Java programs	• It provides a Runtime environment. For Java appln, making Java programs platform-independent.	• It contains & developments tools like the Java Compiler debugger & other.
• Java compiler (Javac): converts Java source code into bytecode.	• It translate. bytecode to machine code	2) contains set of libraries & other files
• JRE: Includes.	• this translation	

The jvm (standard) libaries which is used for Running Java Program.

enables Java appln to run on any devices os. that has comp.

- The JDK is used for developers & provide all the tools necessary to create & manage Java appln, including those required for Running them.

Q.5 What is the Role of JVM in Java? How does the JVM execute Java code?

→ Role:

The JVM enables Java platform independence by abstracting the execution of Java bytecode. From the underlying OS.

- The JVM loads Java bytecode, verifies it & executes it using either an ~~interpreter~~ JIT compiler, converting bytecode to machine code at Runtime.

Q6. Explain the memory management system of the JVM?

→ JVM Has 3 types of memory management.

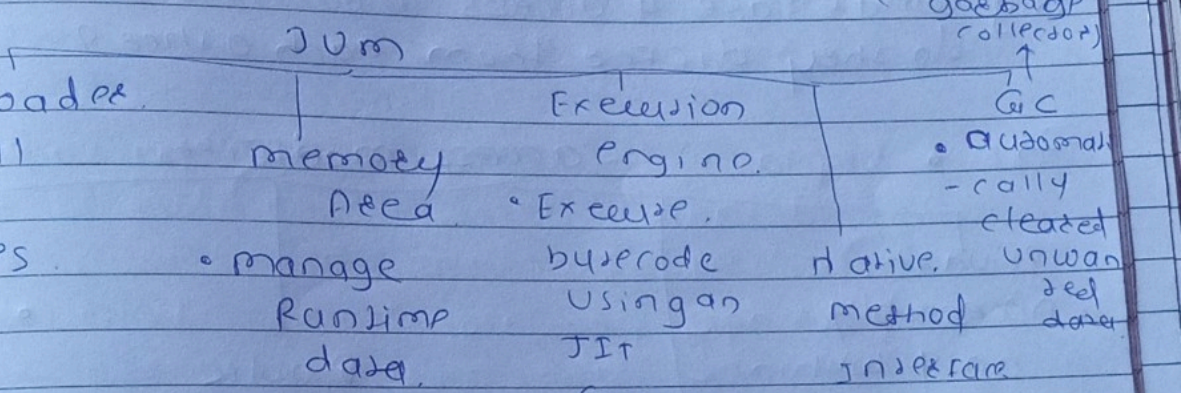
Heap	Stack	Method area
i) store objects & class instance.	• store methods call & local variable.	• store class & execute like Runtime constant & static variables

PC register/store the current address of execution instruction.

Q.5. what are the JIT compiler? ils Role in JVM? what is byte code & why it is imp for java?

→ *JIT Compiler
- Component of JVM that converts bytecode to native machine code at runtime, improving performance.
- Bytecode:
  - An intermediate code generated by java compiler, essential for java's platform independence as it can be executed on any JVM.

Q6 Describe the architecture of JVM
→ etc



Q.7. How does Java achieve platform independance through the Java?
→
- Java achieves platform independence by compiling code into platform.
- A bytecode which in JVM, execute on any platform that has a compatible JVM
- decoupling Java programs from specific OS.

Q.8. What the siginificance of the class loader in Java ? What is the process of gc in Java?

→ i) Class loader:

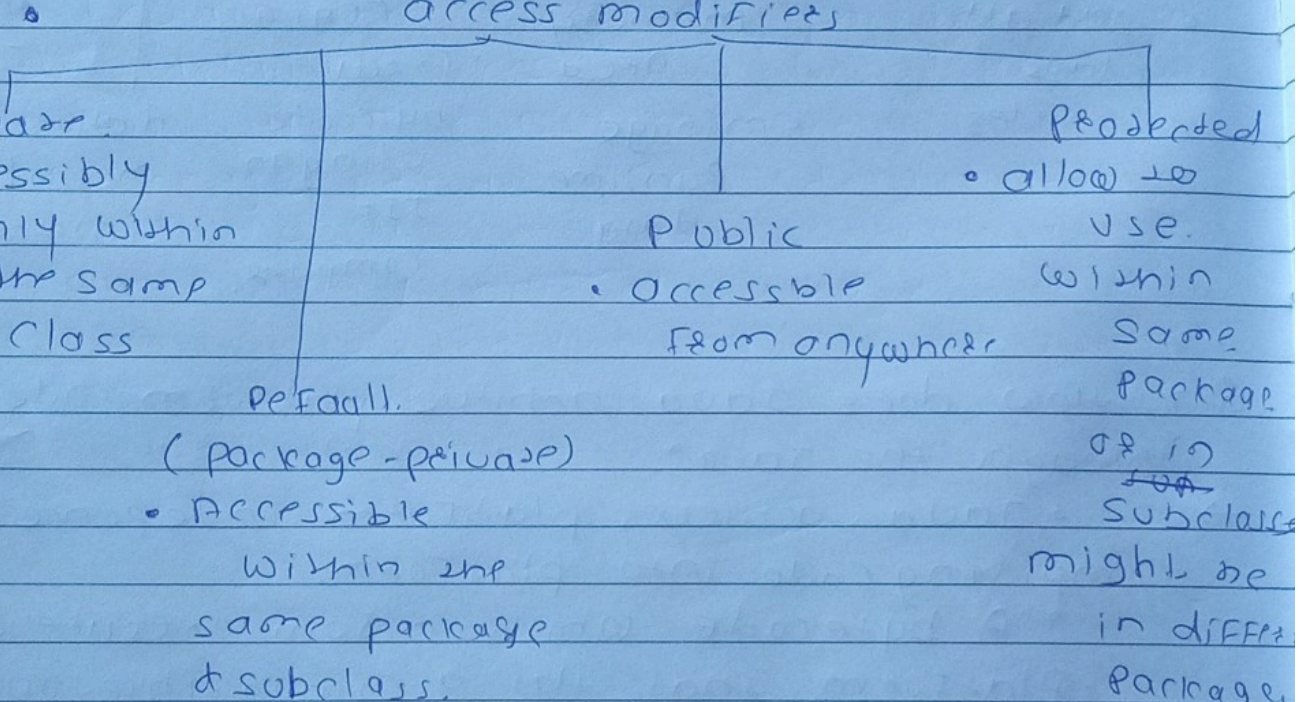
- dynamically load java classes into the JVM at runtime supporting features like dynamic loading & unloading.

2) Garbage collection:

- Automatically frees up memory by destroying Objects that are no longer in use, preventing memory leaks.

Q.9 What are 4 access modifiers in Java & How do they differ from each other ?

→



Q.10. what is difference bet^n public, protected & default ~~class~~ access modifier?

modifier	class	package	sub class	Global
public	✓	✓	✓	✓
protected	✓	✓	✓	×
default	✓	✓	×	×
private	✓	×	×	×

Q.11. Can you override a method with a different access modifiers & in subclass? For example, can a protected method in a superclass be overriden with a private method in a subclass? Within

→ No, we cannot override a method with a ~~stricter~~ access modifier in a subclass.
e.g. a protected method in a superclass cannot be overriden with a private method in a subclass because it would reduce the visibility

Q.12. Is it possible to make a class private in java? If yes, where can it be done & what are the limitations?

→ Yes, we can be private class only if it is [nested class]. A top level cannot be private. as it needs to be accessible at least within its own package

Q.13 Can a Top level class in Java be declared as protected or private? why or why not?

→ • A Top level class in Java can only be public or default (Package-private).
• It cannot declared as protected or private

because those modifiers ~~who~~ would Restrict. the ~~visibility in a way that contra adicts~~ Java package access control.

Q.15 what happens if you declare a variable or method as private in a class & try to access it, From another class within same package.

→ If you declare a variable or methods as in a class, it cannot be accessed From another class within same package. You would need to use public or protected access methods to allows controlled access

Q.16 Explain the concept of "package private" or "default" access. How does it affect the visibility of class members?

→ • package private access (default)-means that the class, methods, variables are accessible only current package.
  • This is the limitation of visiblity of default.
  • It is not accessable in other package