Q1] JDK consist of Java Virtual Machine and other resources like interpreter / loader and compiler an archiver a documentation genrator.

- Java compiler is like a translator that converts human readable code into a machine readable format. It checks the syntax and detects compile time error.
- Java doc it generates documentation for Java code.
- This tools help your package your Java classes and resources into a single archive file, like a zip file, for distribution or deployment.
- Class loader have different sub components which ~~works~~ are responsible for loading bootstrap path, extension path, class loader, custom loader.
- Jvm is heart of the Java which is used to convert byte code to native machine code.
-

Q2]

	JDK	JVM	JRE
-	JAVA development kit	Java Virtual Machine	Java Runtime Envoirment.
-	Java development kit is a tool used to develop Java application	Java Virtual Machine converts byte code to the native machine code	it's software bundle used to run java code.
-	JDK is platform dependent	Jvm is platform independent	JRE is also platfor dependent.
-	Superset of JRE	It is the subset of JRE	it is subset of JDK.

Q3] JVM is mainly responsible for 3 activites

- Loading
- Linking
- Initialization.

JVM Byte code execution

-loading:

The Class loader reads the ".class" file, generate the corresponding binary data and save it in the method area.

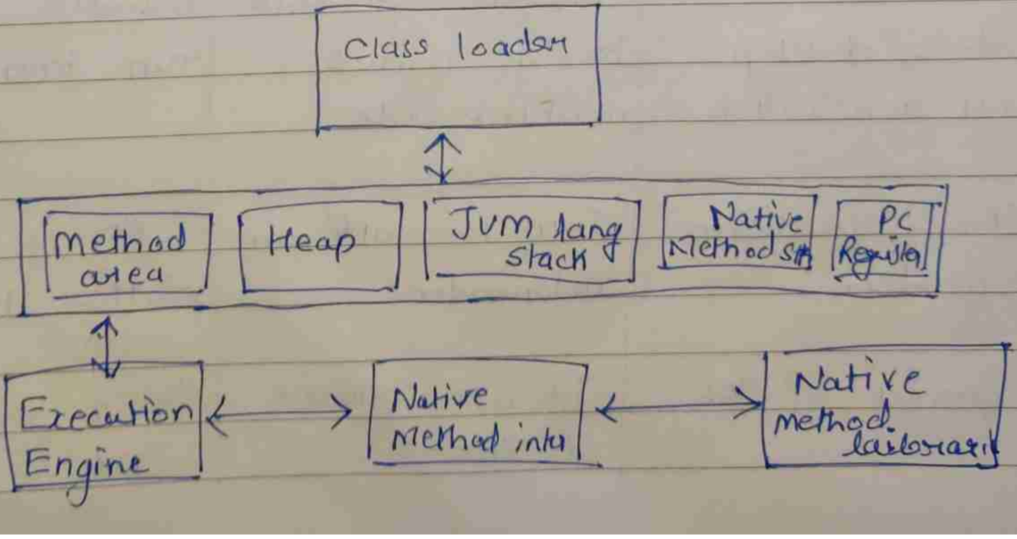
- After loading the ".class" file, JVM creates an object of type class to represent this file in the heap memory. Please note that this object is of type class predefined in java.lang. pacakage

-linking:

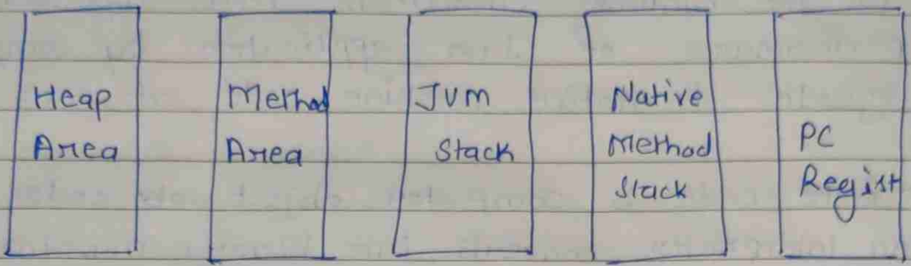
linking performs verification, preparation, and resolution.

- Initialization:

all static variables are assigned with their values defined in the code and static block.



Q4] Explain Memory management System of JVM?



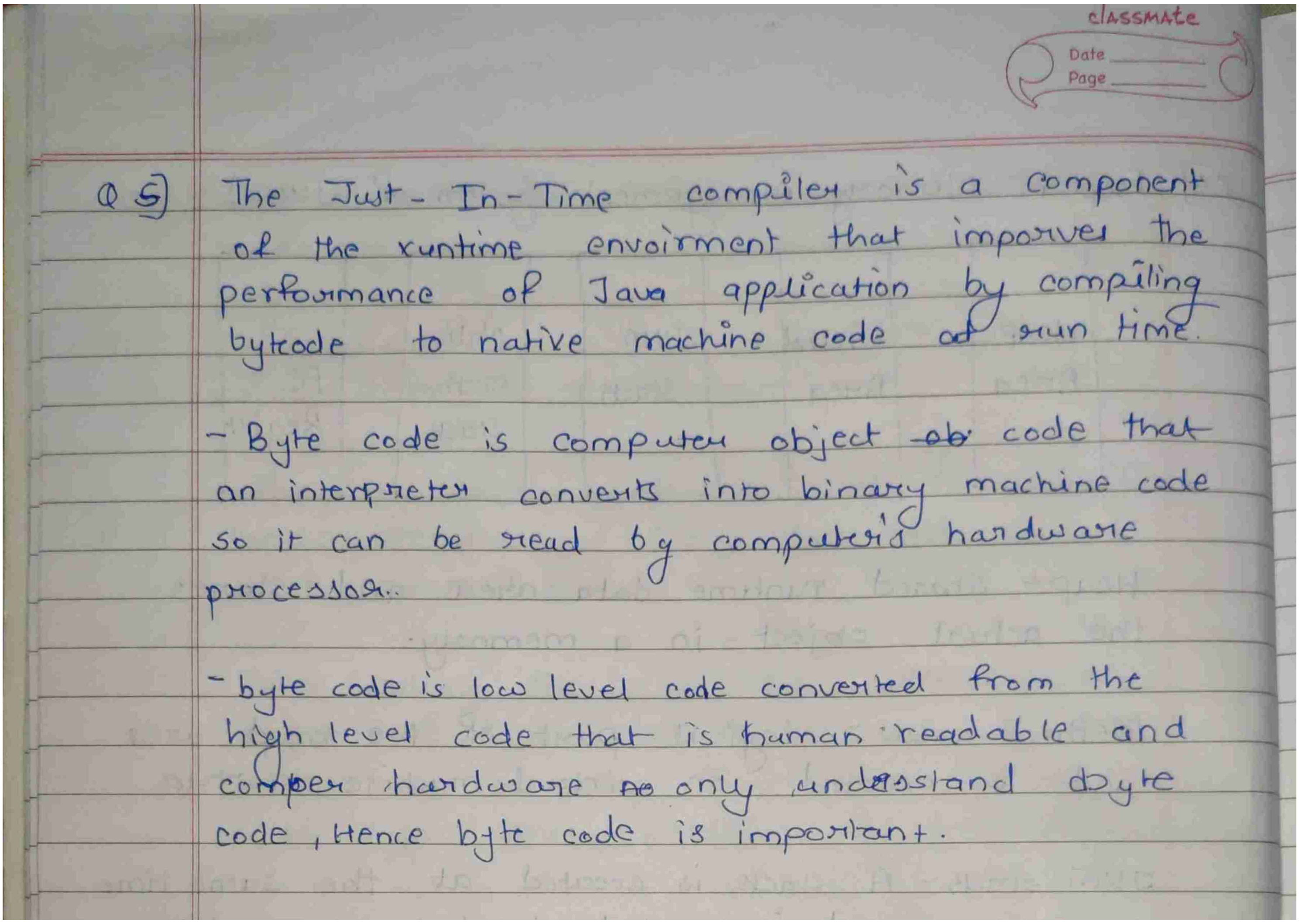
Heap - Shared runtime data area and stores the actual object in a memory.

Method area - logical part of the heap area and is created on virtual machine startup.

JVM stacks - A stack is created at the same time when the thread is created. Used to store data and partial result which will be need to while returning the value for method and performing dynamic linking.

Native stack - Also called has c stack, native method stacks are not written in Java language This memory is allocated for each thread when it's created.

Program counter - PC register is capable of storing the return address or a native pointer on some specific platform.

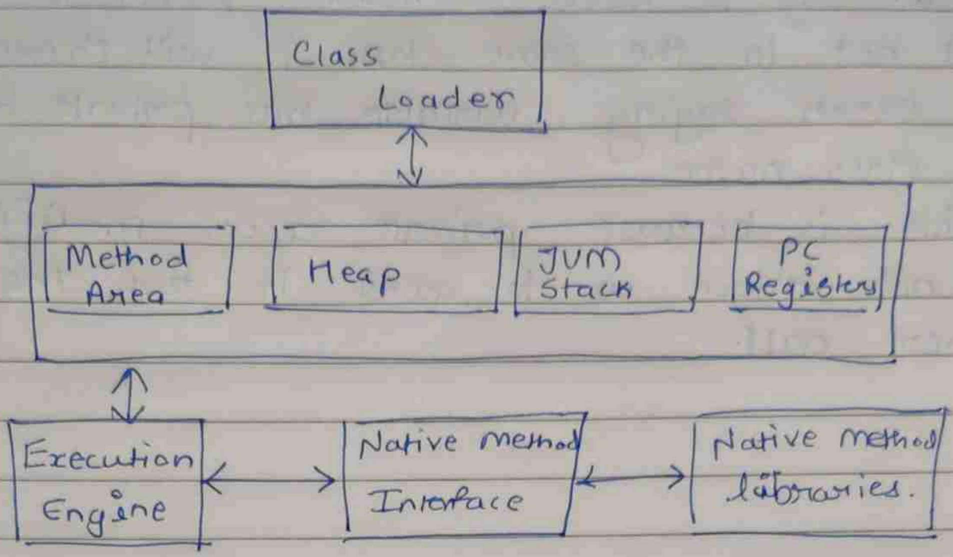


Q 5) The Just-In-Time compiler is a component of the runtime envoirment that imporves the performance of Java application by compiling bytcode to native machine code ~~at~~ run time.

- Byte code is computer object ~~ob~~ code that an interpreter converts into binary machine code so it can be read by computer's hardware processor..

- byte code is low level code converted from the high level code that is human readable and comper hardware ~~no~~ only understand byte code, Hence byte code is important.

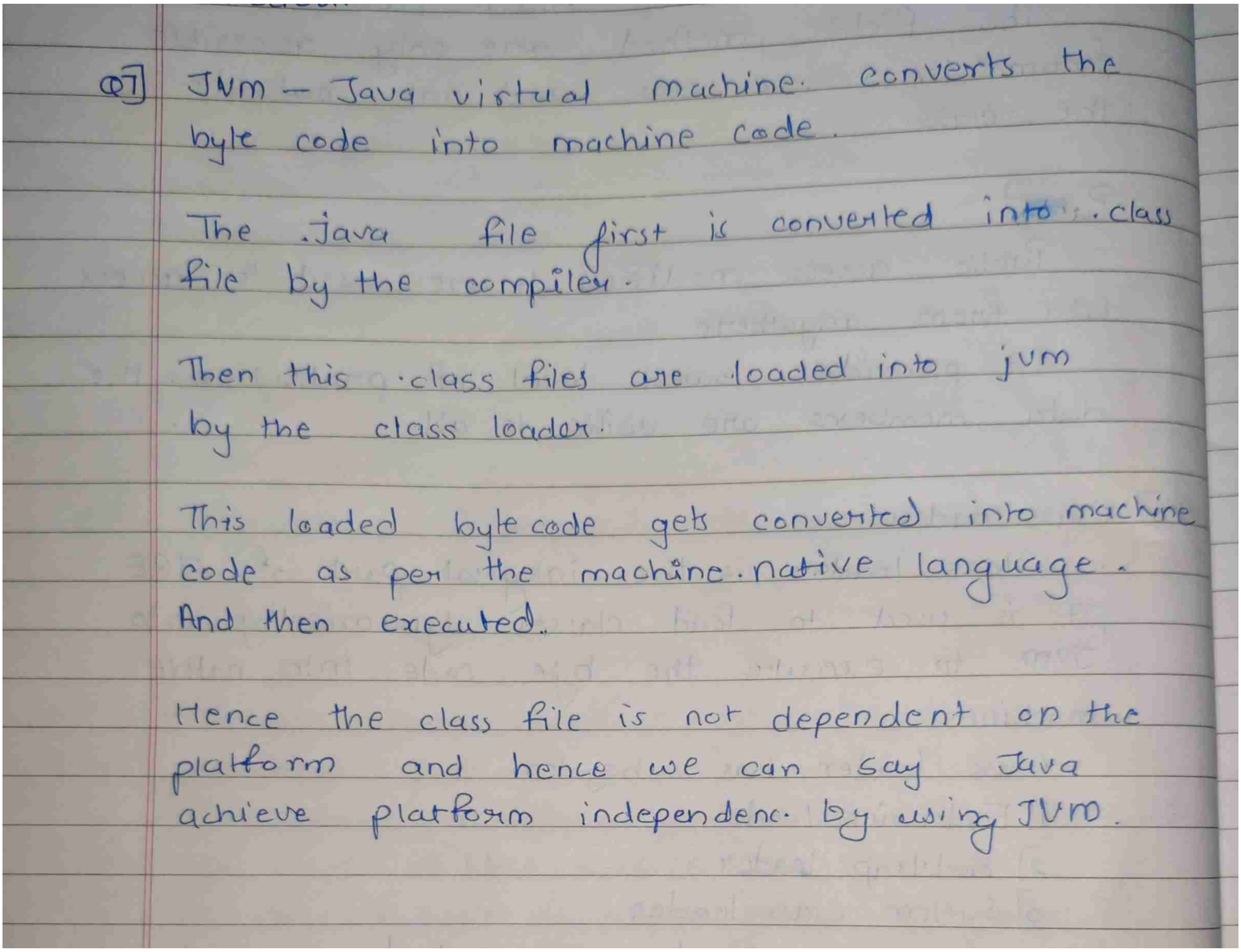
Q6]



- Class loader:-
  - loading
  - linking
  - Initialization
  - Verification

- Class (Method) Area:
  Class (method) Area stores per-class structures such as the runtime constant pool, field and method data, the code for methods
- Heap- it is runtime data area
- Stack - hold local variables .
- Program ~~Con~~ Counter Register:
  Java virtual machine instruction currently being executed.
- Native method Stack:
  It contains all the native methods.

- JNI (Java Native interface):
  it's an API used to communicate with another application written in another language c, c++;



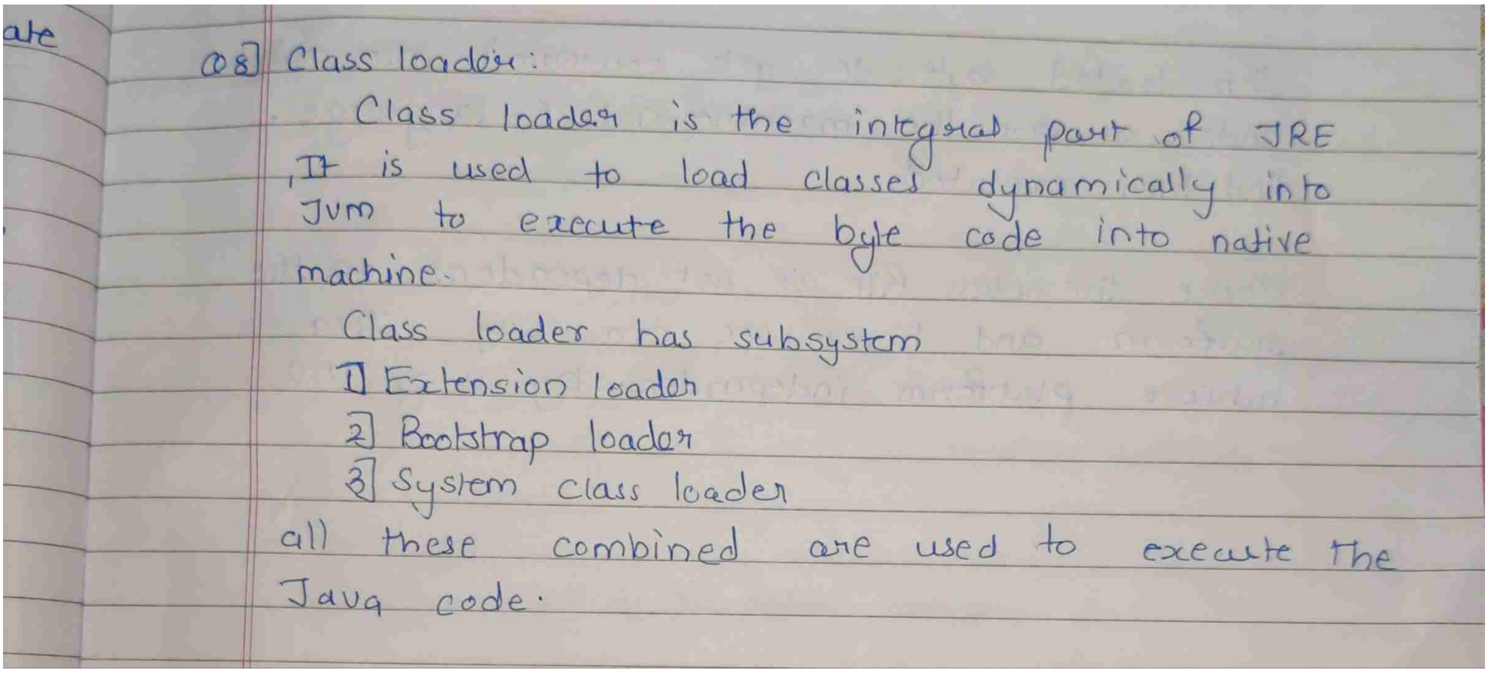
Q7] JVM - Java virtual machine. converts the byte code into machine code.

The .java file first is converted into .class file by the compiler.

Then this .class files are loaded into jvm by the class loader.

This loaded byte code gets converted into machine code as per the machine. native language. And then executed.

Hence the class file is not dependent on the platform and hence we can say Java achieve platform independenc. by using JVM.



ate

Q8] Class loader:

Class loader is the integral part of JRE ,It is used to load classes dynamically into Jvm to execute the byte code into native machine.

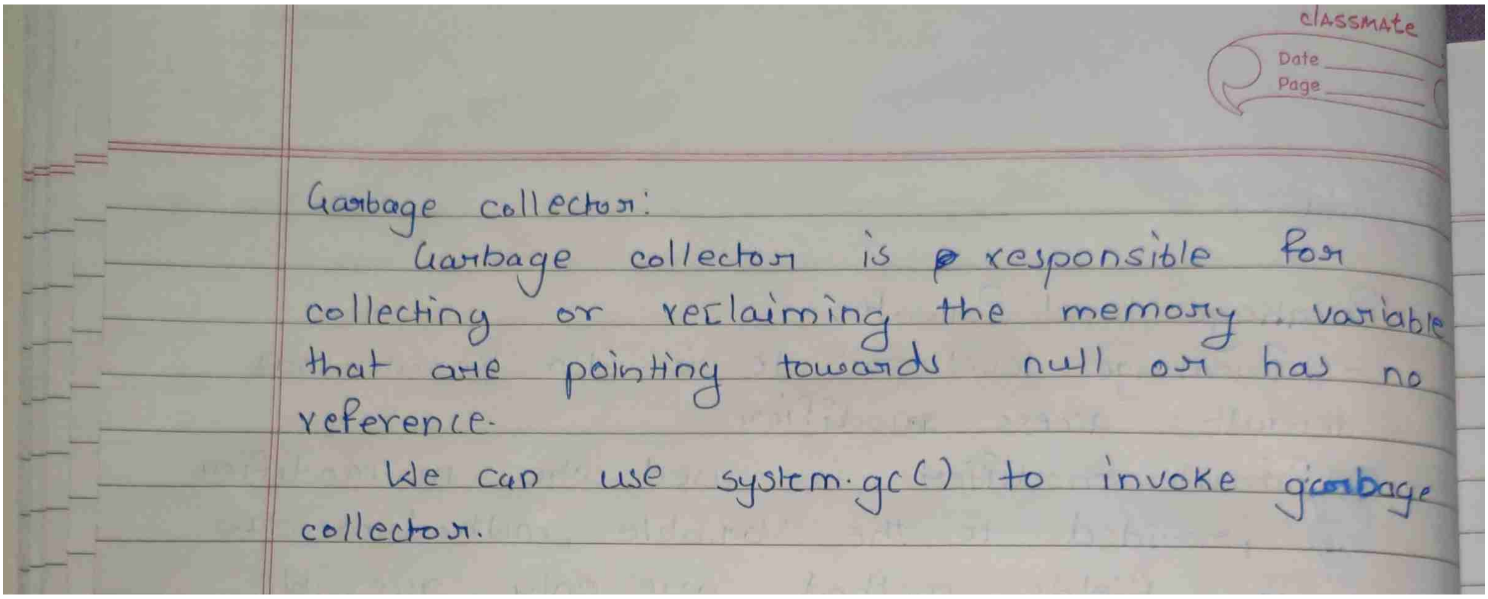
Class loader has subsystem

1] Extension loader

2] Bootstrap loader

3] System class loader

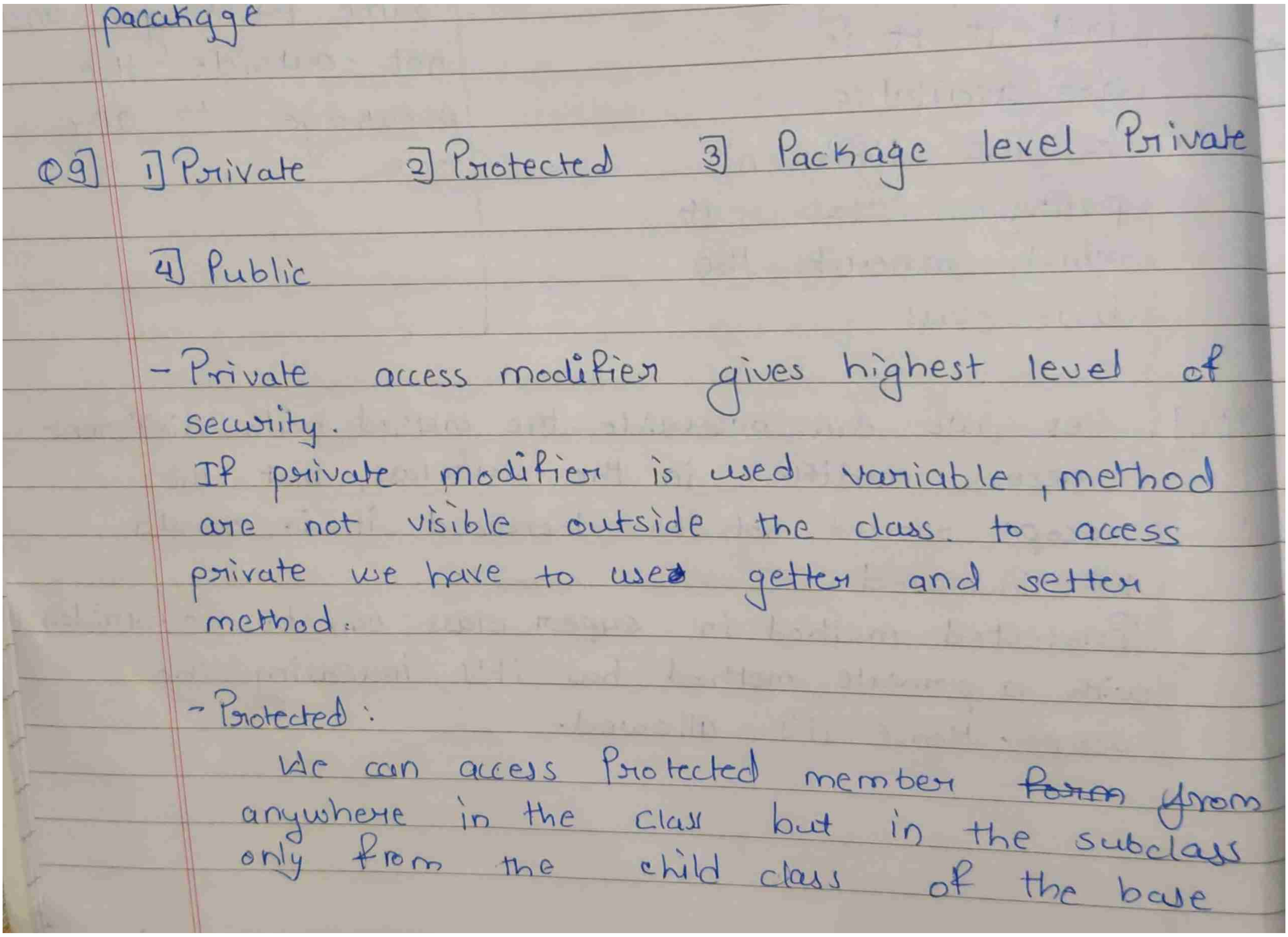
all these combined are used to execute the Java code.



Garbage collector:

Garbage collector is responsible for collecting or reclaiming the memory variable that are pointing towards null or has no reference.

We can use system.gc() to invoke garbage collector.



pacakgge

Q9] 1] Private 2] Protected 3] Package level Private

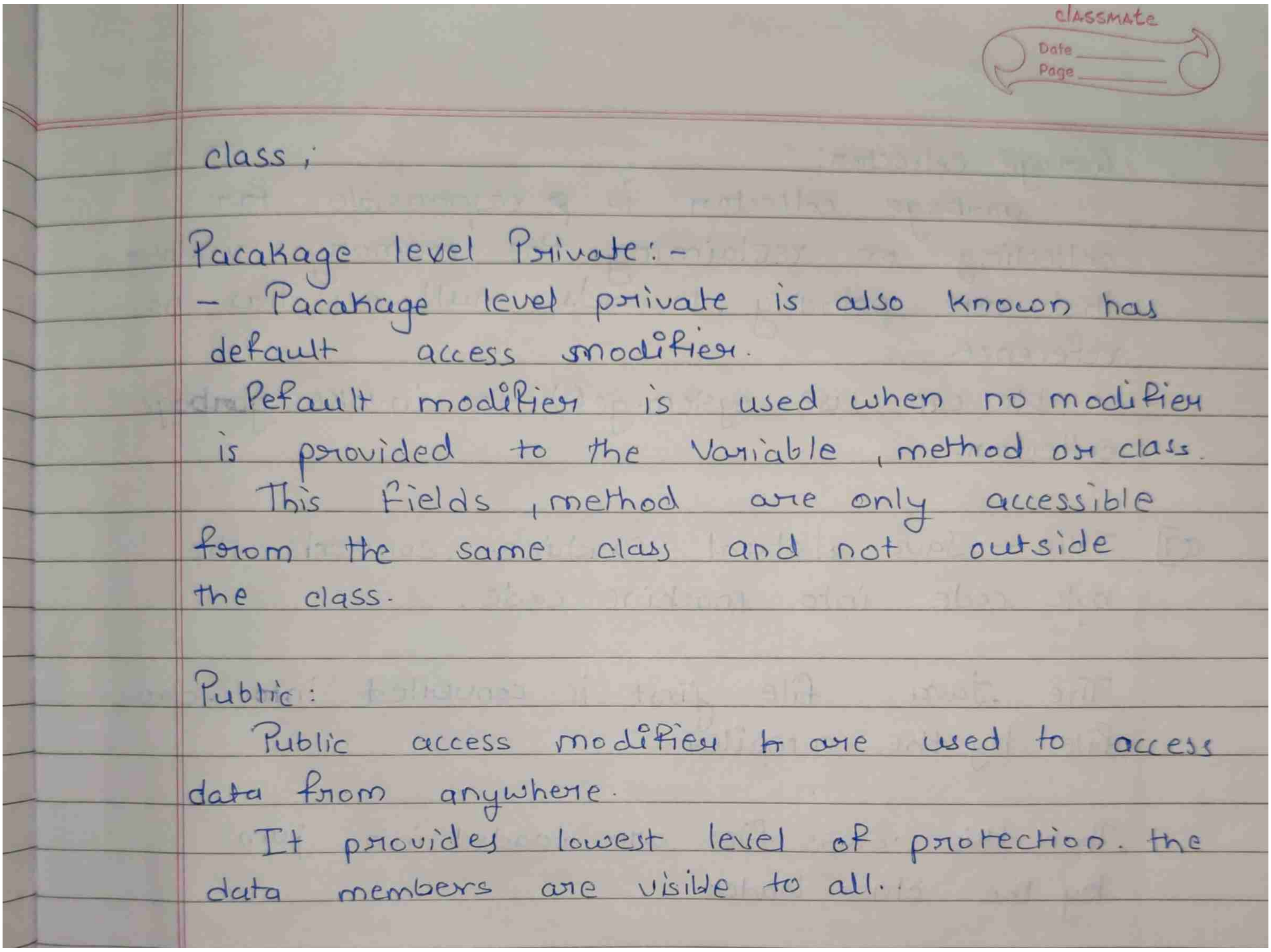
4] Public

- Private access modifier gives highest level of security.

If private modifier is used variable, method are not visible outside the class. to access private we have to used getter and setter method.

- Protected :

We can access Protected member ~~form~~ from anywhere in the class but in the subclass only from the child class of the base



class ;

Pacakage level Private :-

- Pacakage level private is also known has default access modifier.

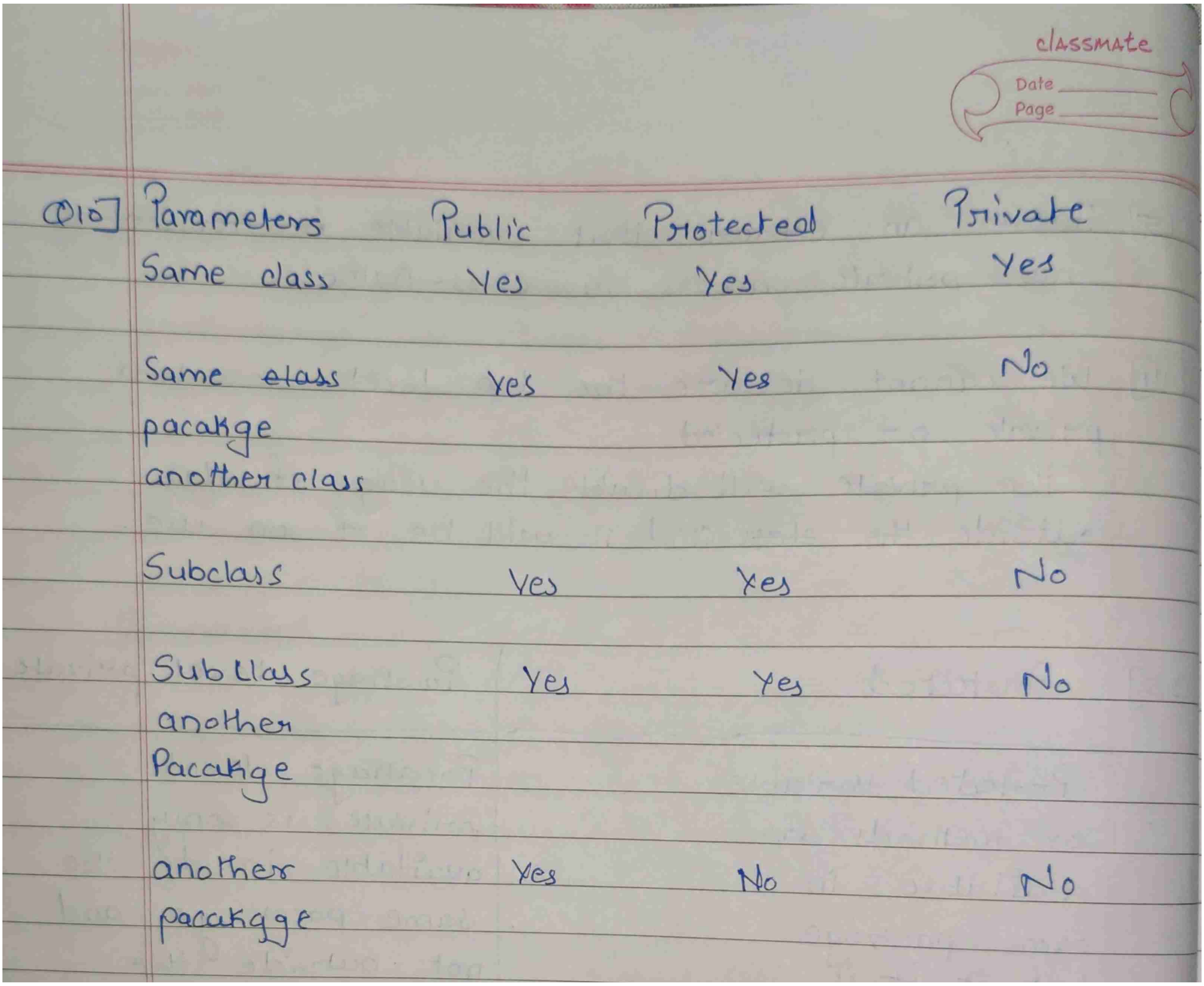
Default modifier is used when no modifier is provided to the Variable, method or class.

This fields, method are only accessible from the same class and not outside the class.

Public :

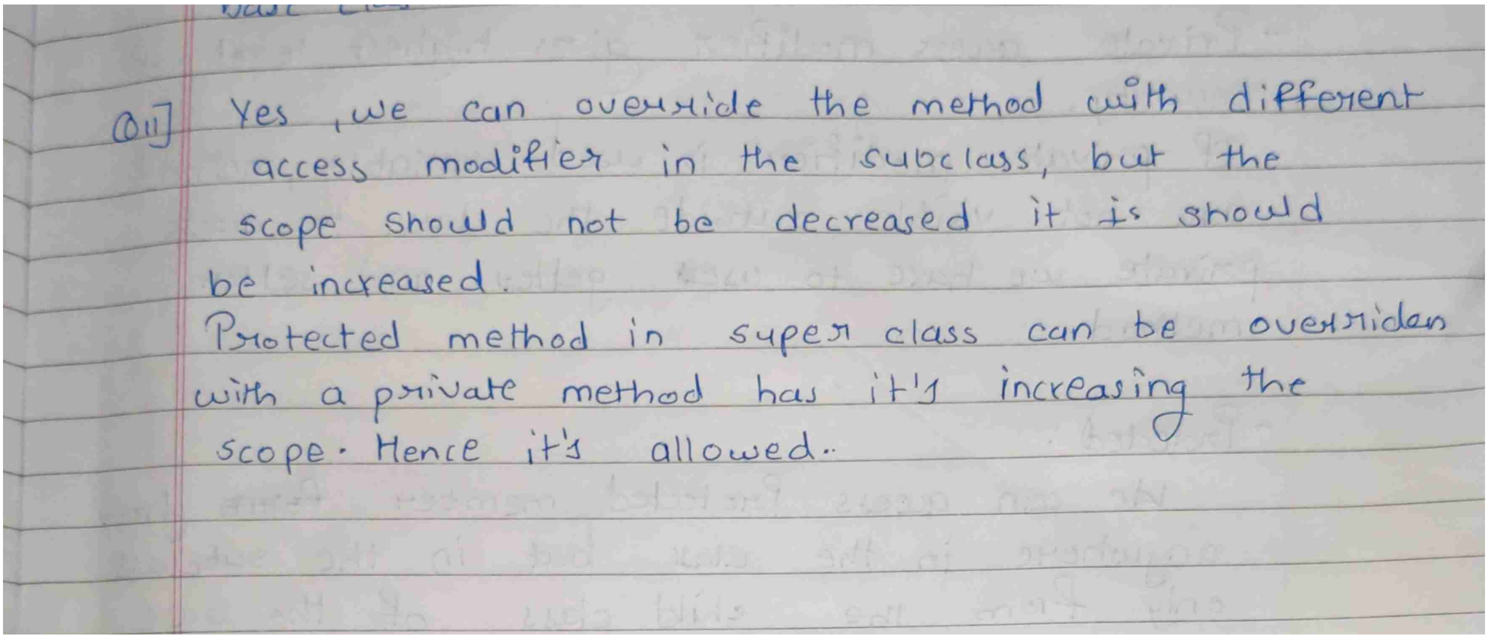
Public access modifier ~~to~~ are used to access data from anywhere.

It provides lowest level of protection. the data members are visible to all.



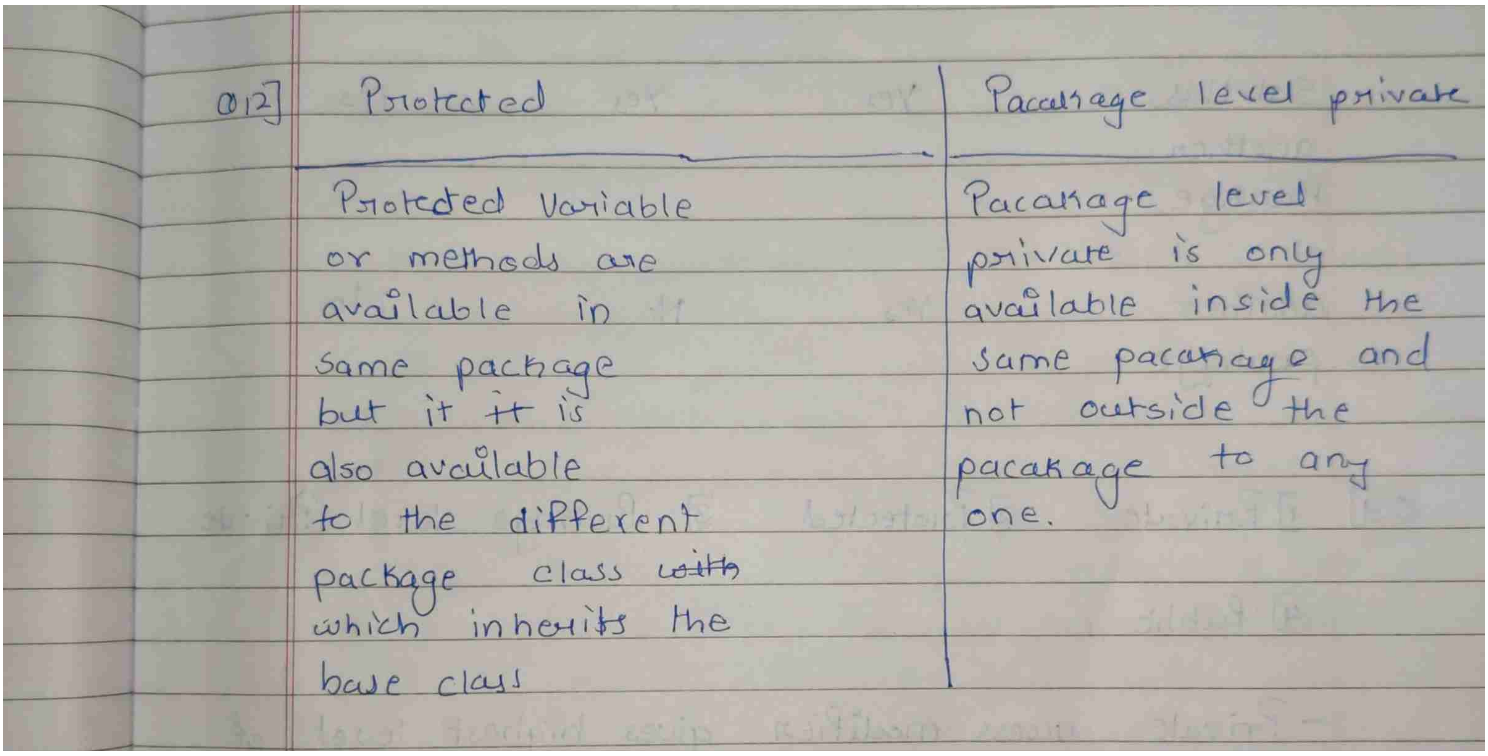
Q10]

Parameters	Public	Protected	Private
Same class	Yes	Yes	Yes
Same ~~class~~ pacakge another class	Yes	Yes	No
Subclass	Yes	Yes	No
Sub class another Pacakge	Yes	Yes	No
another pacakgge	Yes	No	No



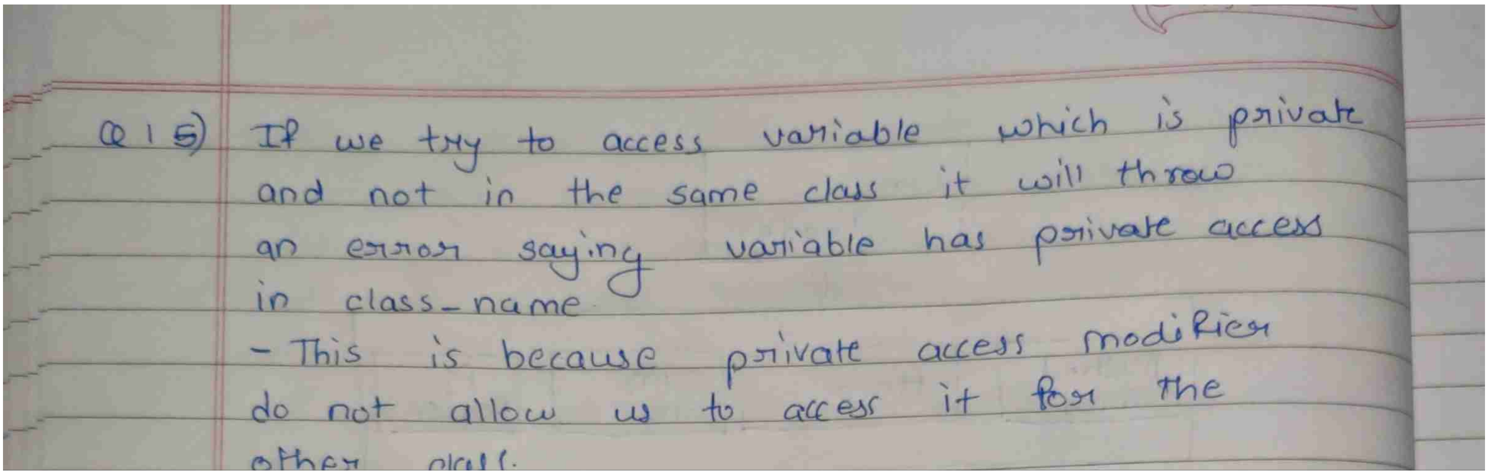
Q11] Yes , we can override the method with different access modifier in the subclass, but the scope should not be decreased it is should be increased.

Protected method in super class can be overriden with a private method has it's increasing the scope. Hence it's allowed..



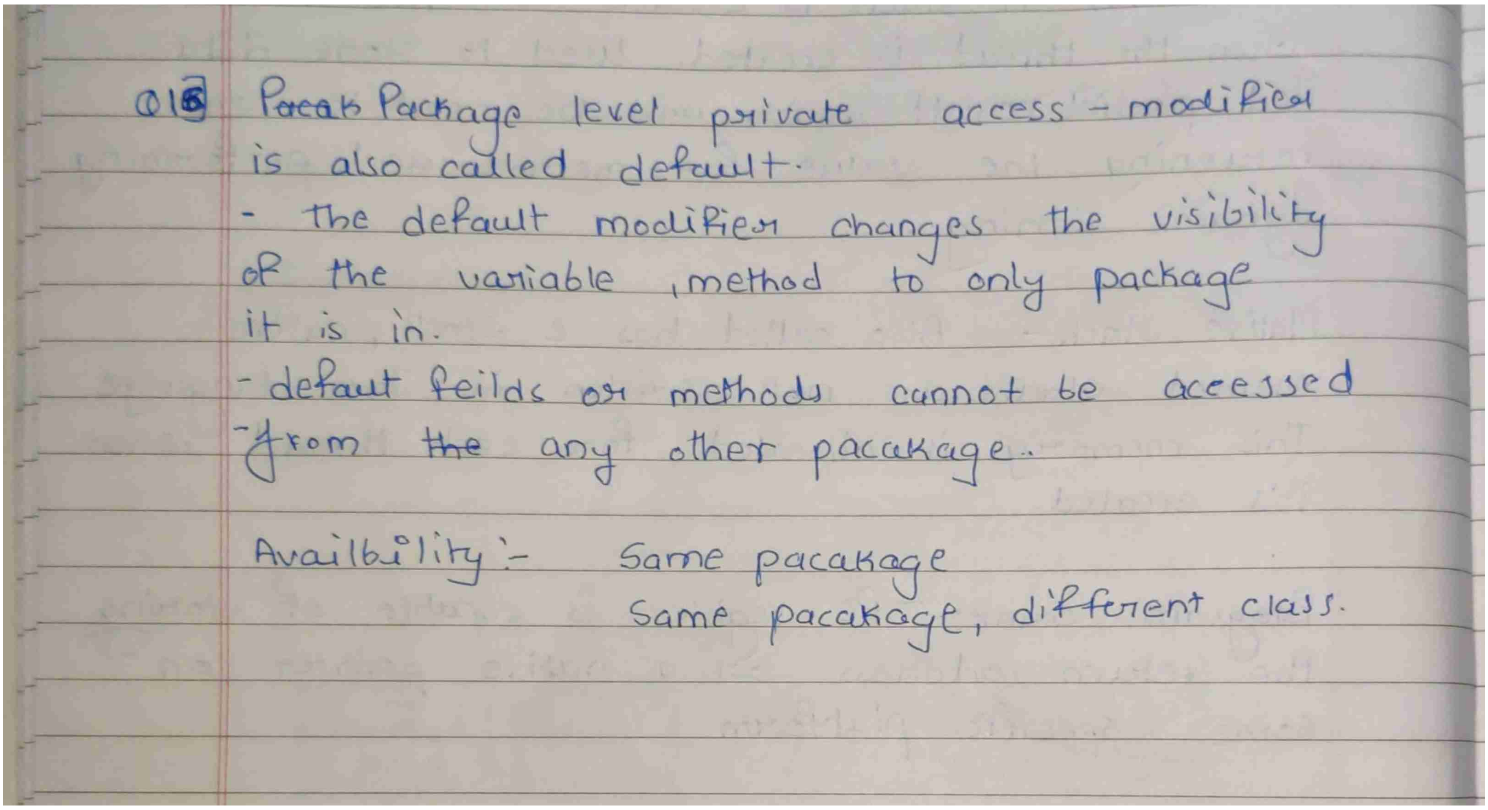
Q12]

Protected	Pacakage level private
Protected Variable or methods are available in same package but it ~~it~~ is also available to the different package class ~~with~~ which inherits the base class	Pacakage level private is only available inside the same pacakage and not outside the pacakage to any one.



Q15) If we try to access variable which is private and not in the same class it will throw an error saying variable has private access in class_name

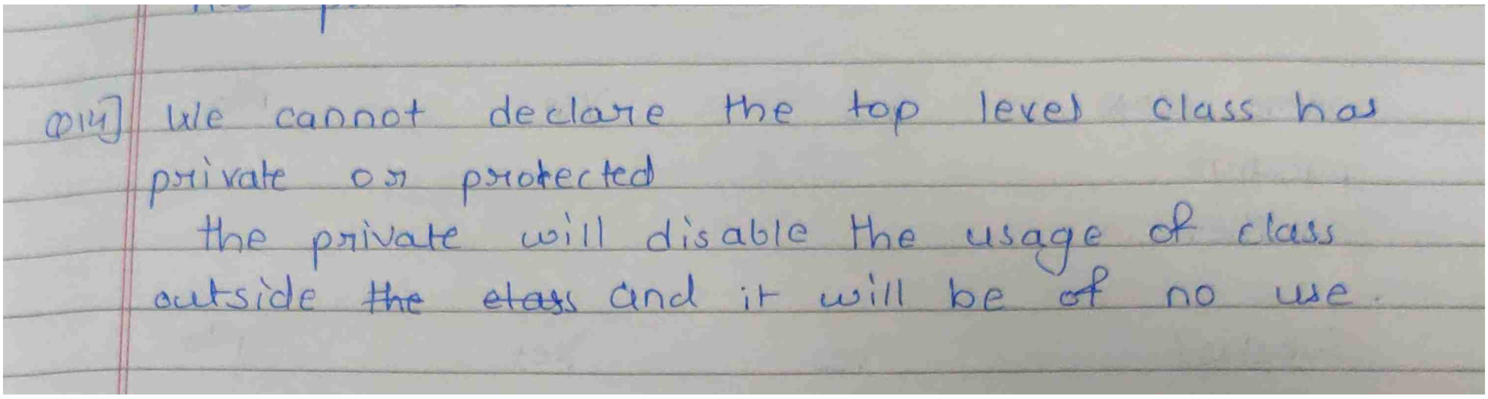
- This is because private access modifier do not allow us to access it for the other class.



Q15 ~~Pacak~~ Package level private access - modifier is also called default.

- The default modifier changes the visibility of the variable ,method to only package it is in.
- default feilds or methods cannot be accessed from ~~the~~ any other pacakage.

Availbility :- Same pacakage
Same pacakage, different class.



Q14] We cannot declare the top level class has private or protected

the private will disable the usage of class outside ~~the~~ ~~class~~ and it will be of no use.