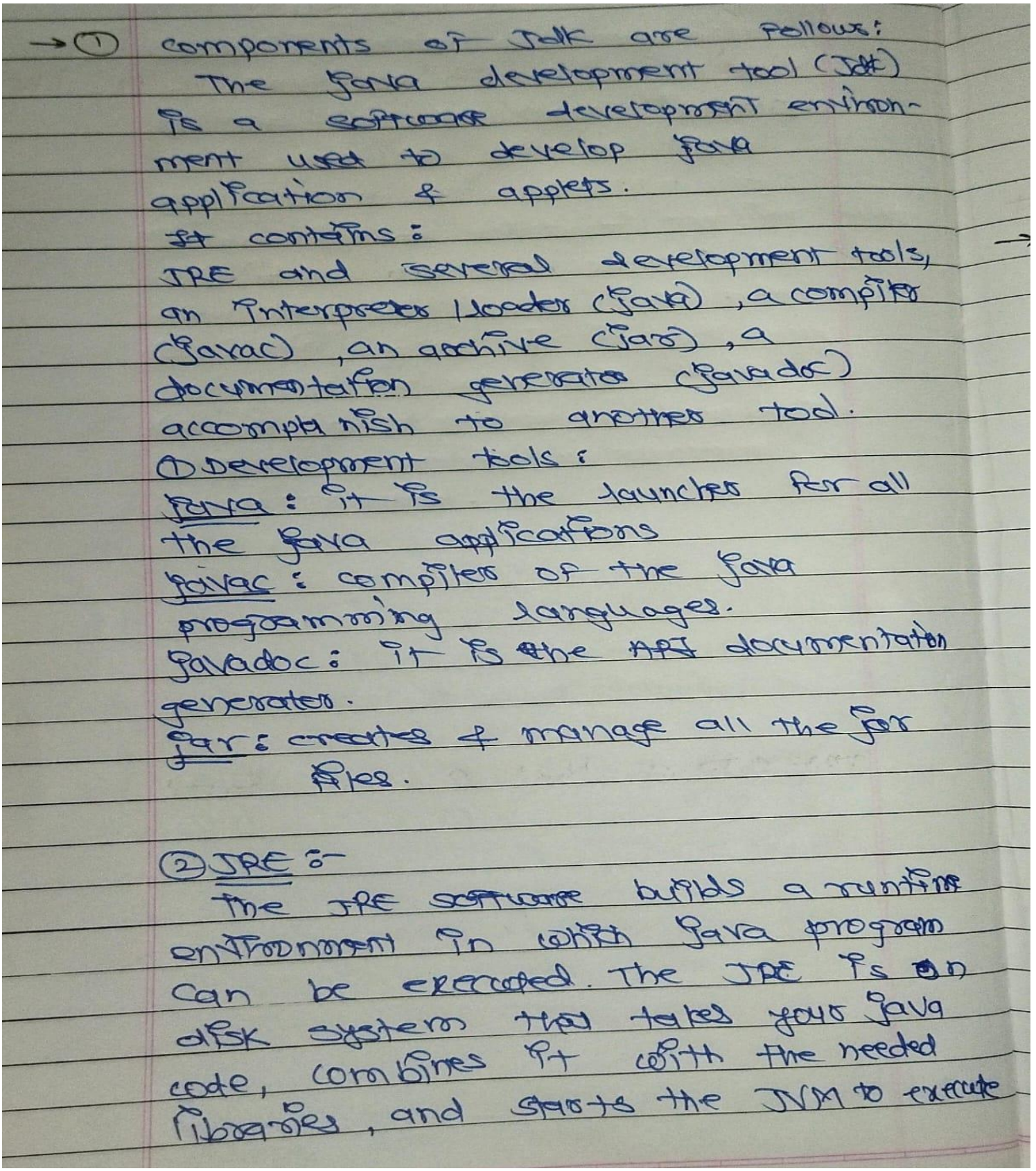
Java Assignment 02

1) Explain the components of the JDK.

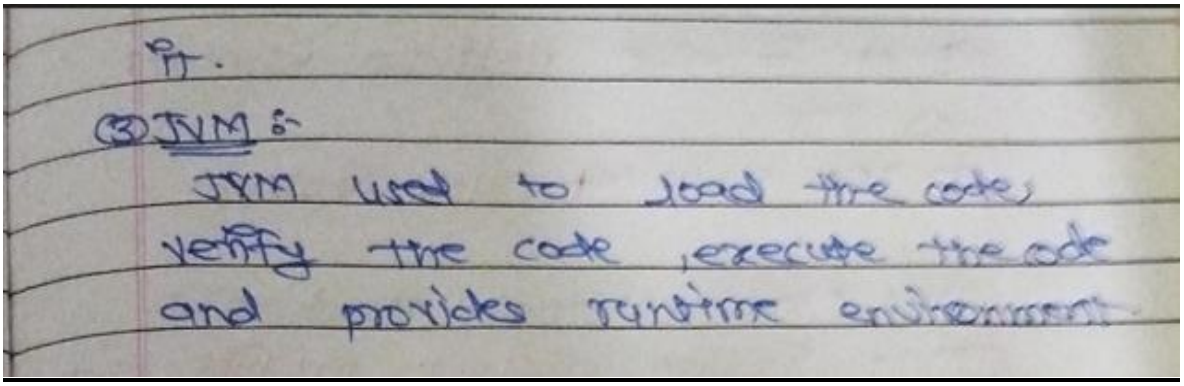
=>



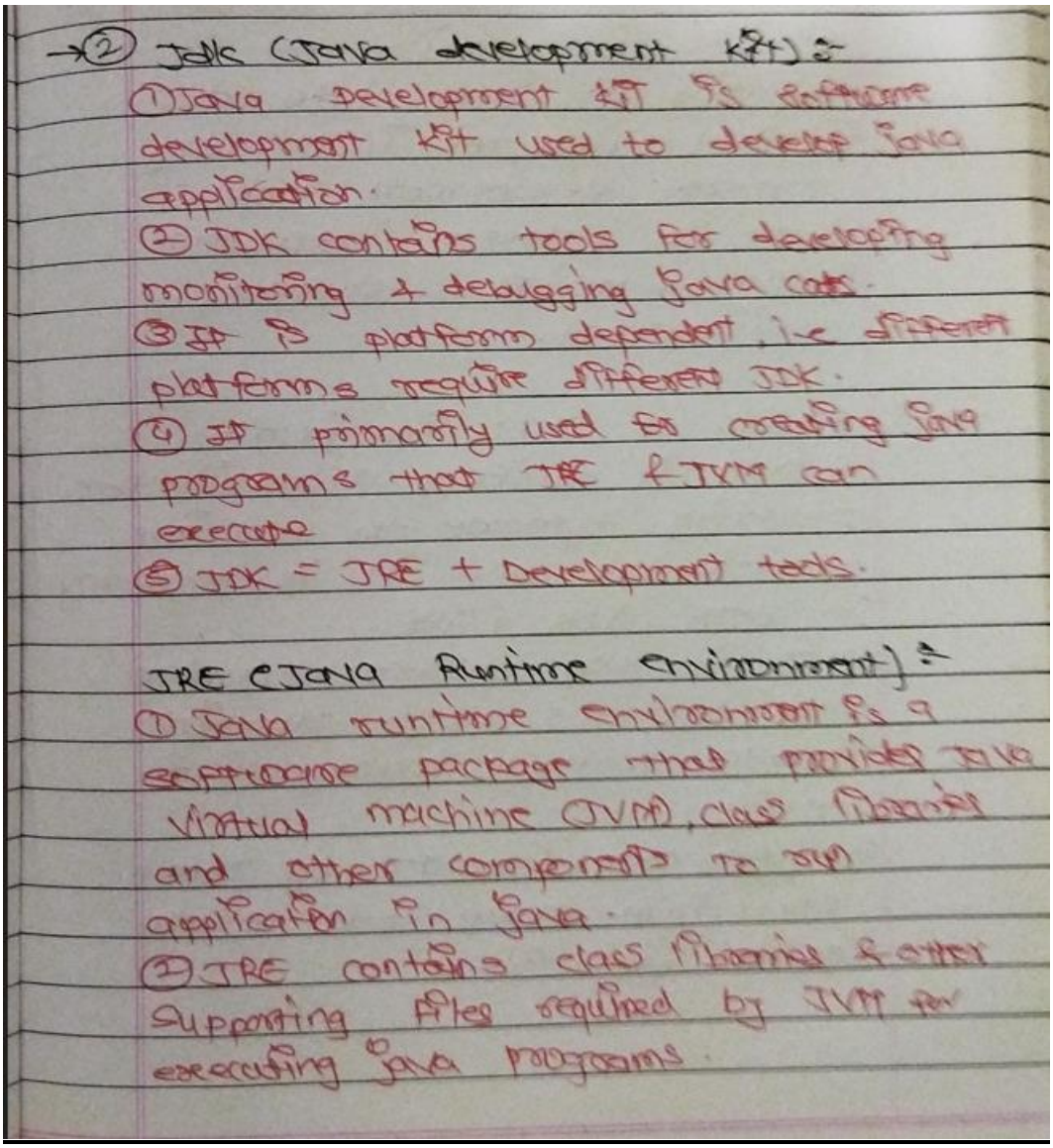
① components of JDK are follows:
The java development tool (JDK) is a software development environment used to develop java application & applets.
It contains:
JRE and several development tools, an Interpreter/loader (java), a compiler (javac), an archive (jar), a documentation generator (javadoc) accomplish to another tool.

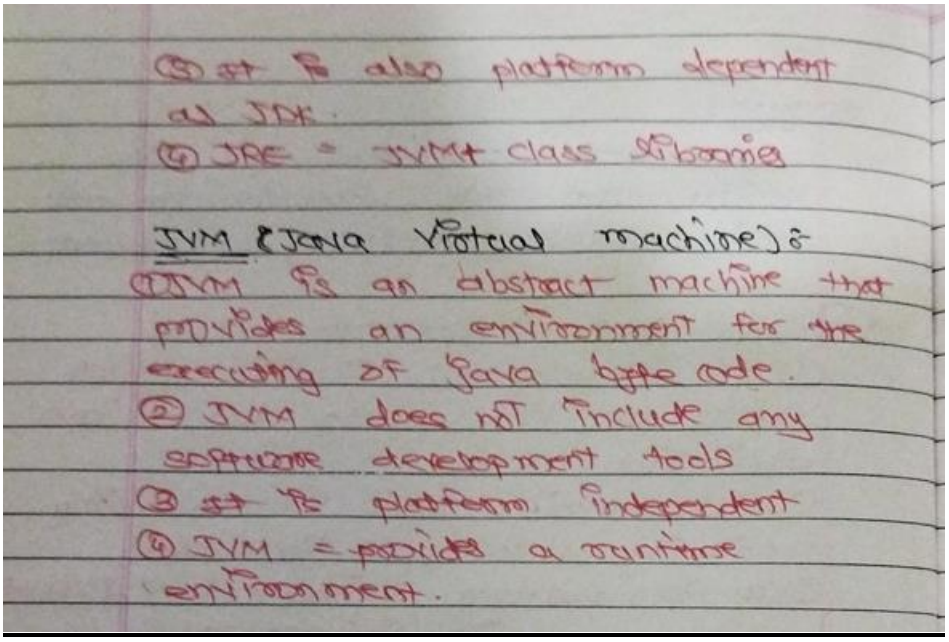
① Development tools:
java: it is the launcher for all the java applications
javac: compiler of the java programming languages.
javadoc: it is the API documentation generator.
jar: creates & manage all the jar files.

② JRE:
The JRE software builds a runtime environment in which java program can be executed. The JRE is on disk system that takes your java code, combines it with the needed libraries, and starts the JVM to execute

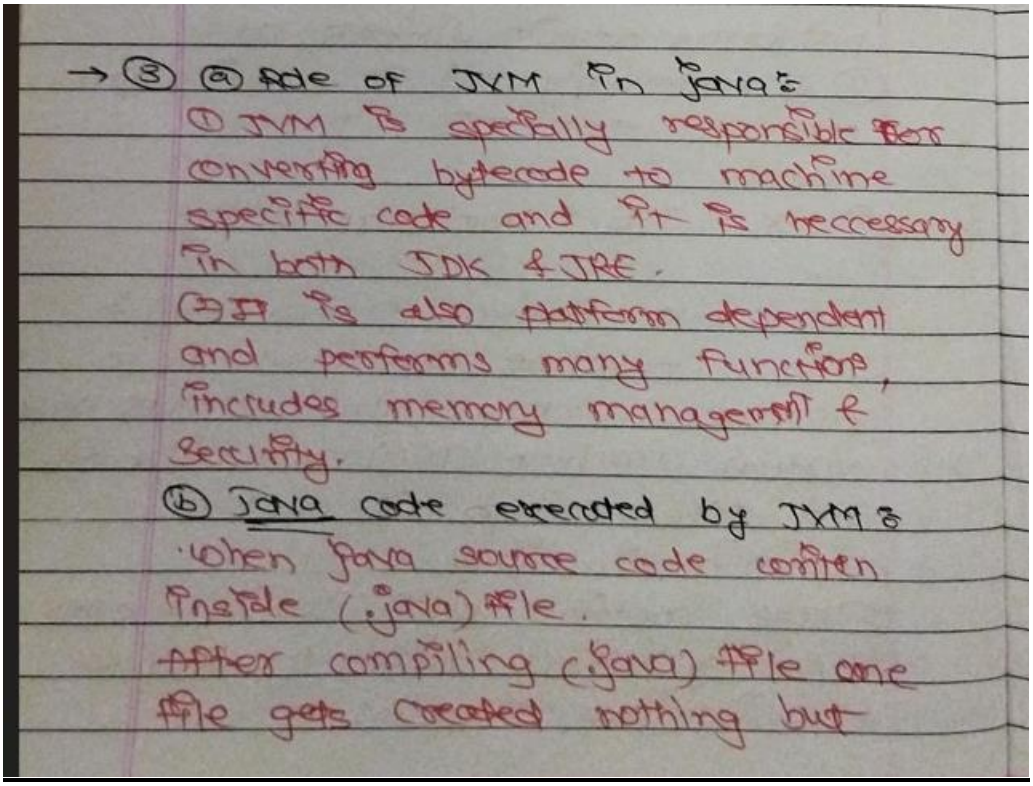


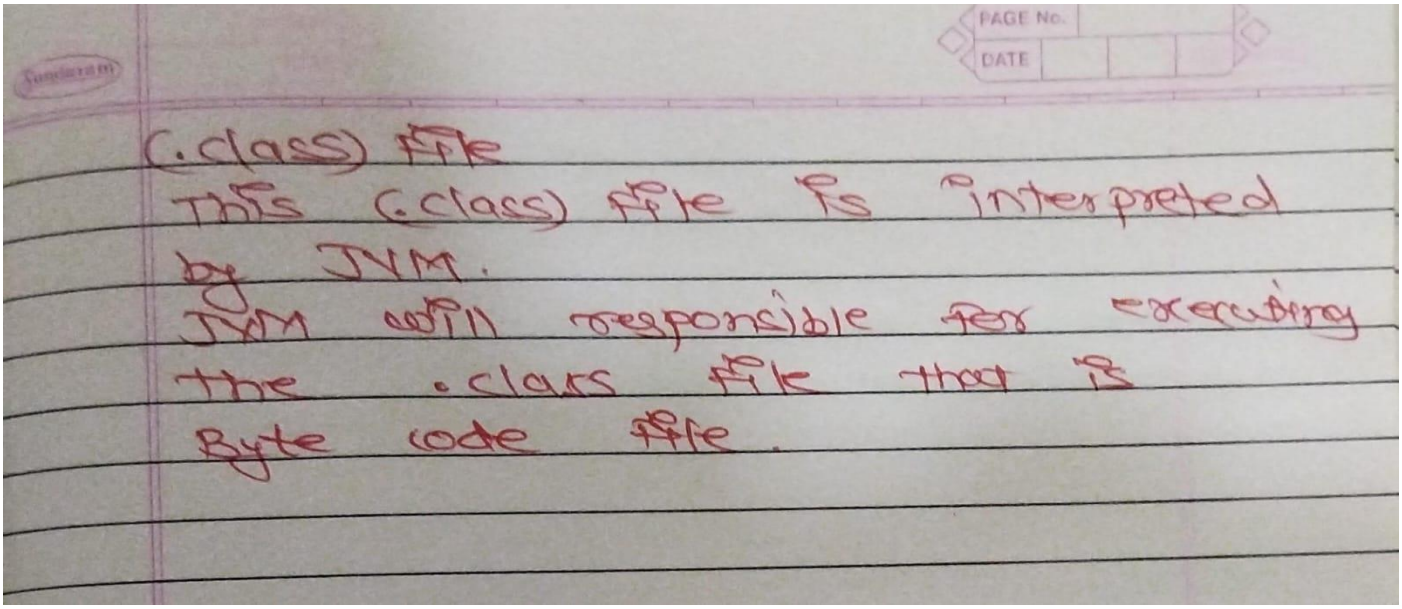
2) Differentiate between JDK, JVM, and JRE.





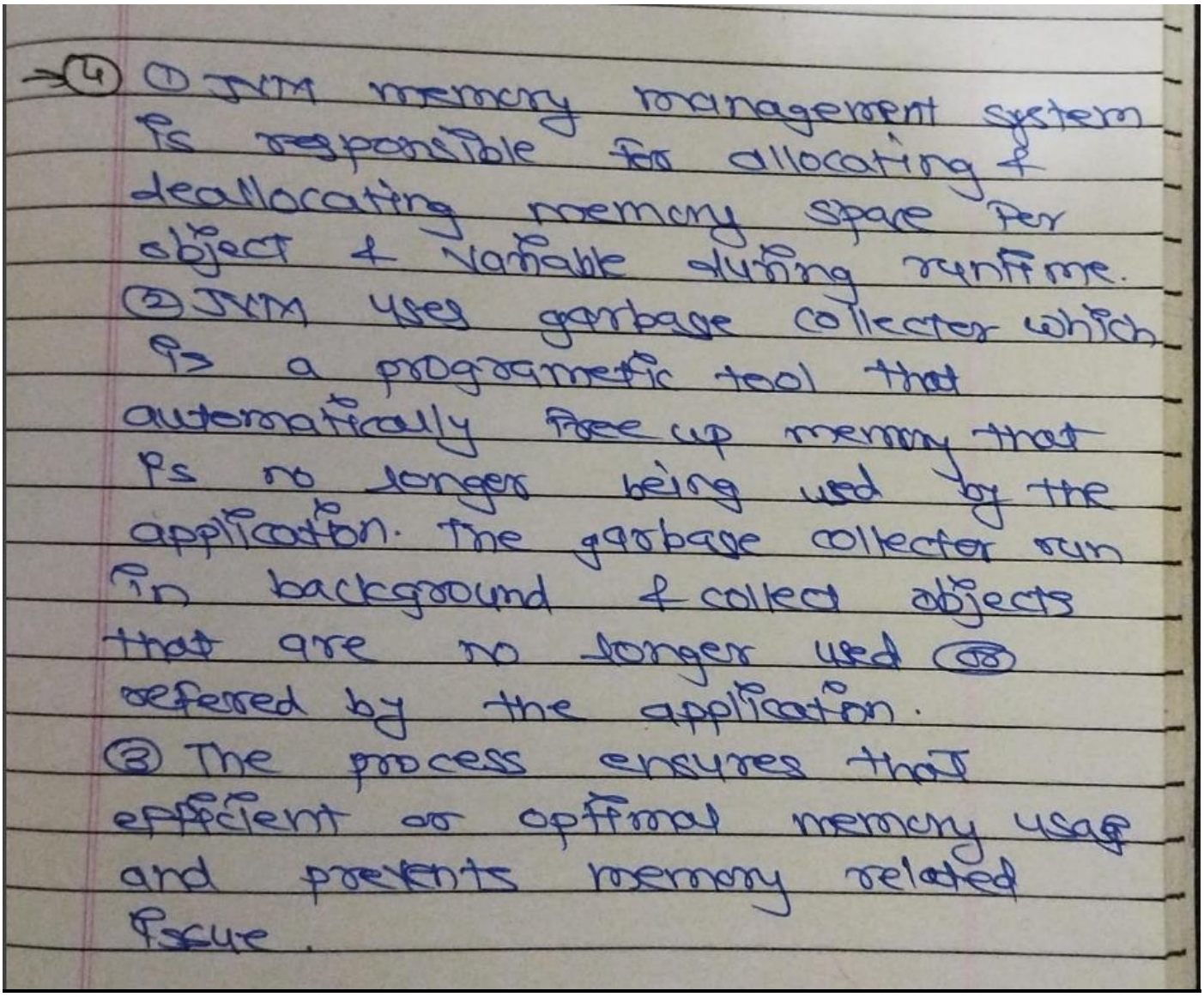
3) What is the role of the JVM in Java? & How does the JVM execute Java code?





4) Explain the memory management system of the JVM.

=>



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

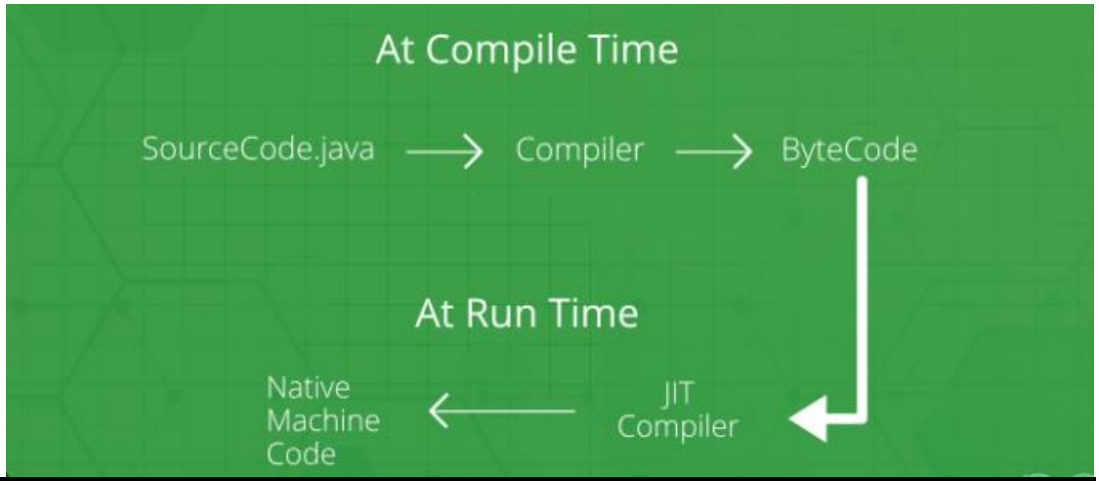
=>

1) JIT in java is na integral art of JVM.

2) It accelerates execution performance many times a long running, computer intensive Program that provides the best performance og java application at compile or Run time.

3) The JIT compilation includes two approaches AOT(Ahead-of-Time) and interpretation to translate code into machine code.

4) AOT compiler compile the code into a native machine language.



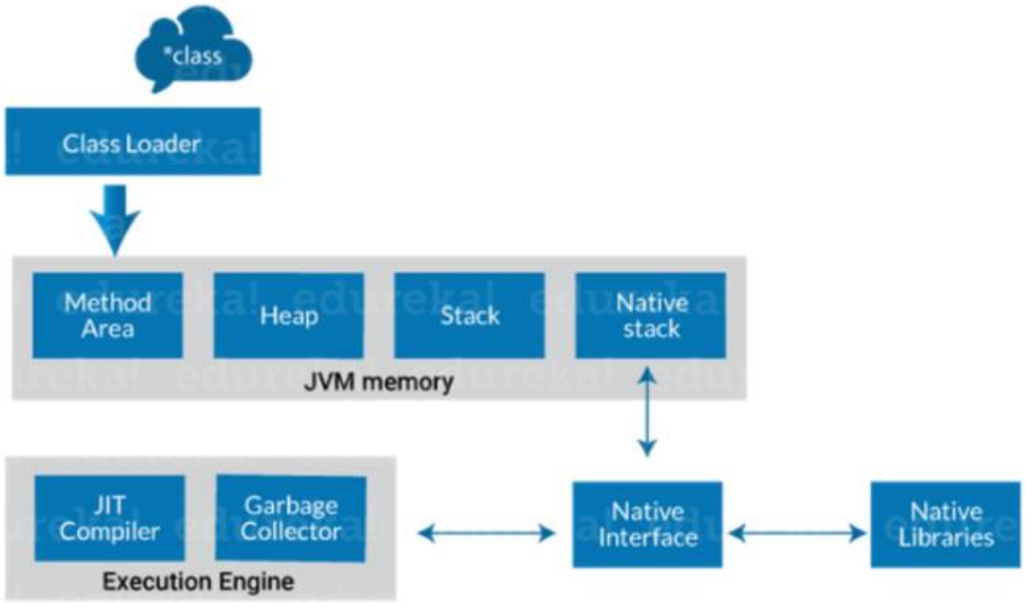
5) **Byte Code**:

Java Byte Code is instruction set of Java Virtual Machine(JVM) for executing program written in java language and other JVM compatible language.

Each byte code operation in the JVM is represented by a single.

6) Describe the architecture of the JVM.

=>



1) **Class Loader**:

-> Class loader is a subsystem of JVM. It is used to load class files.

-> Whenever we run the java program, class loader loads it first.

2) Class method area:

-> It is one of the Data Area in JVM, in which Class data will be stored.

-> Static Variables, Static Blocks, Static Methods, Instance Methods

are stored in this area.

3) Heap:

-> A heap is created when the JVM starts up. It may increase or decrease

in size while the application runs.

4) Stack:

-> JVM stack is known as a thread stack.

-> It is a data area in the JVM memory which is created for a single execution thread.

-> The JVM stack of a thread is used by the thread to store various elements i.e. local variables.

5) Native stack:

-> It subsumes all the native methods used in your application.

6) Execution Engine:

-> JIT Compiler

-> Garbage Collector

7) How does Java achieve platform independence through the JVM?

=>

1) Java achieved platform independence through the Java Virtual Machine(JVM),

Which serves as an abstraction layer between the compiled Java Code and

The underlying hardware and operating System.

- Byte Code
- Just-in-Time(JIT)
- Class Libraries
- Portability

2) Java achieved platform independence by compiling source code into byte code Which is executed by the JVM and dynamically translated into native machine Code, allowing to run reamlessly across diverse platforms.

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

=>

1) Class Loader are responsible for loading Java classes dynamically to the JVM during runtime.

2) They are also art of JRE, therefore the JVM doesn't need to know about the underlying files or file system in order to run java programs. They are responsible loading classes into memory.

3) **Garbage Collector:**

Garbage Collector in java is the automated Process of deleting code that is no longer need or used.

This automatically free up memory space and ideally make Java apps easier for Developers.