Assignmen 2.

1. Explain the component of JDK.

→ JDK is known as Java development kit. We have to understand that the tools required for developer are development tool, ~~score~~ source code their documentation & supporting library. The compounding of the mentioned tools is known as SDK, which is software development kit

- SDK :-

1. Development tools - Tools to develop
2. Documentation - documentation of tools.
3. library - supporting files to develop.
4. Runtime Environment - To test the code.

Now along the same structure JDK components can be shown as :-

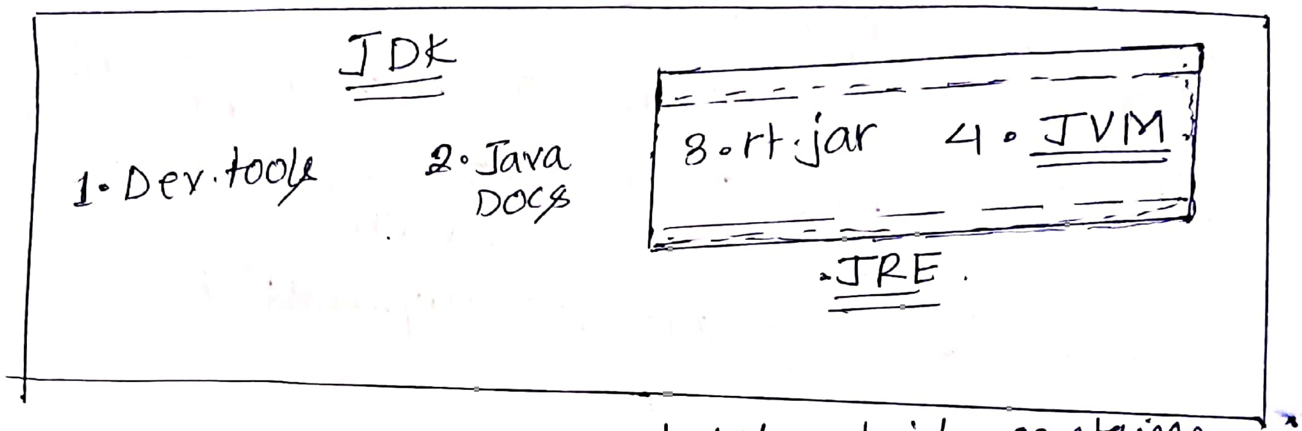
1. Dev. tools → Java development tools } Helps to develop application.
2. Documentation → Java API Docs.
3. library → rt.jar. } JRE.
4. RE → Java Runtime Environment }

∴ We can say components of JDK is Java dev tools, Java API docs & JRE.

2. Differentiate between JDK, JVM and JRE.

→ JDK – Java development kit.
JVM – Java Virtual M/C.
JRE – Java Runtime Environment.

With the help of diagram the classification of JDK, JVM, JRE is as follows.



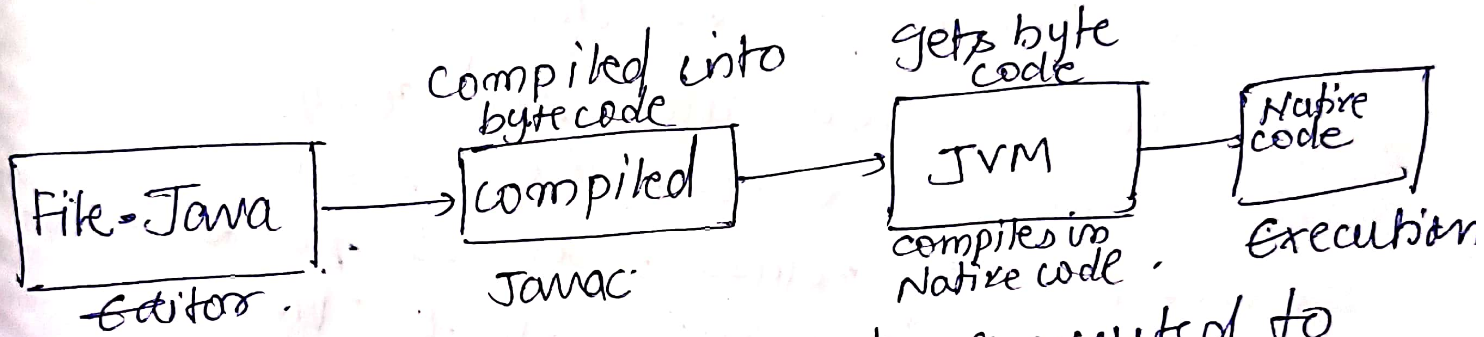
1. JDK is a development kit which contains all the tools required for a developer to develop the Java application.

2. JRE is known as java runtime environment which contains the libraries as well as JVM. This is necessary for a java program to run on a device.

3. JVM is Java Virtual M/C which is subset of JRE. It is what helps run the byte code which is compiled by java compiler.

3. What is the role of the JVM in Java? How does JVM execute Java Code?

→ JVM is basically the virtual Machine which gets the compiled code of JAVA through Java compiler, checks all the simentics within code & execute it by generating connection with operating system & run the Machine code.



When compile file.java gets converted to the file.class file which is executable by JVM.

4. Explain memory management of JVM.

→ Basically JVM M.Management is divided into four components

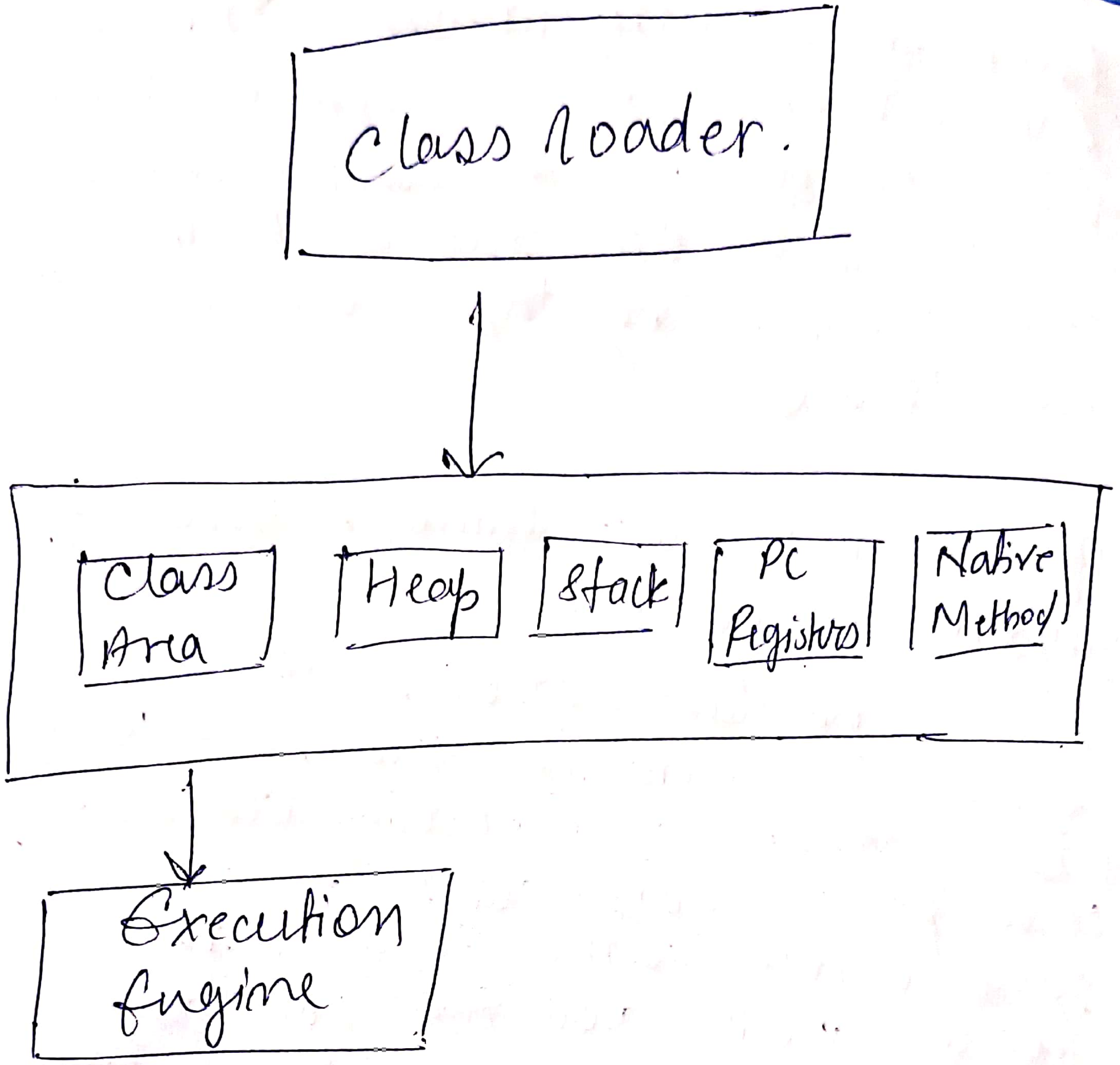
1. JVM memory structure
   - → 1.1 Heap
   - → Method
   - → JVM stack
   - → Native code
   - → PC Registers.

~~2. [illegible]~~

5. JIT & its role in JVM? what is bytecode..

→ J.I.T stands for Just in time, the name itself relates to the time optimisation. JIT is used to compile bytecode to native machine code to increase efficiency at run time. It gives JVM pre compiled method at the time of JVM compilation hence increasing efficiency.
Bytecode is basically the compiled code by Java compiler having extension of .class, it is portable in nature & with the help of JVM it is possible to execute it into native code across the platform.

6. Architecture of JVM.
→ Java virtual Machine basically provides the Environment for java program to test for errors.
The Architecture of JVM is as follows.



7. How does Java achieve platform independence through JVM?

→ As the JAVA team made the slogam write once run anywhere was because of JVM. So the process of Java code is like it gets compiled into Bytecode & JVM is responsible for executing JAVA code to native code so even if we have bytecode with the help of JVM we can run the compiled code on any device hence same compilation time on every other device which will also prevent from data leakage or would avoid the problem of changing syntax according to the device.

8. Significance of class loader & Process of Garbage Collection.

→ Java class loader is a ~~ab~~ class which is present in Java.lang package. It is responsible for loading classes. it helps to load classes. Classes have binary name so a class loader locates data that constitutes a definition for class & if needed helps to provide it in Runtime.

Garbage Collector :- Garbage collector is responsible to release heap memory by eliminating inaccessible objects. It operates in background which is also the feature of Java. It basically is automated in Java which scans the heap memory & distinguish b/w used & unused objects & remove the unused & free memory