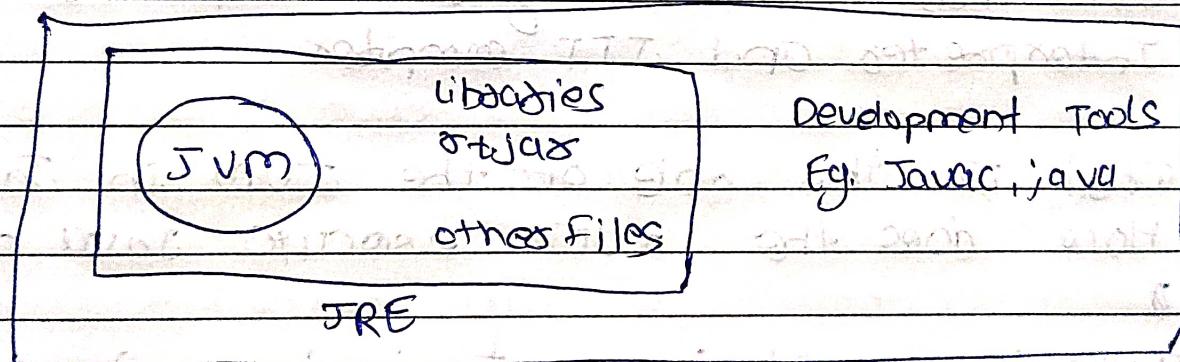


Assignment NO:3

- Q) Explain the components of the JDK?
- ⇒ JDK contains several essential tools and resources for developing applications using the Java programming language.
 - JDK contains a Java virtual machine (JVM) and other few resources such as
 - ⇒ JDK contains JRE and inside JRE there is JVM and In JDK there is other API's
 - ⇒ API are in JDK like `java.lang`, `java.util`, `java.io`, `java.sql` etc. API are present in JDR.
 - JRE is the Java runtime environment. It is also in JDK. And inside JRE there is JVM.

2) Differentiate b/w JDK and JVM and JRE?

2) What are the differences between JDK and JRE?





JDK

- ① JDK is a Software development kit for Java
- JDK = java development tools + Java API's + Java Supporting lib + Java execution environment.
 - JDK contains = java development tools + Java API tools + JRE (rt.jar + JVM)

JRE

- Run Java program on developer's machine/client - we need JRE
- JRE comes with JDK
- JRE contains :- rt.jar + JVM
- rt.jar contains all .class file of our libraries

JVM

- It has 3 section class loader subsystem, runtime data areas and execution engine
- JVM actually convert our byte code to native code
- JVM has execution engine in that there is Interpreter and JIT compiler

Q3) What is the role of the JVM in Java?

How does the JVM execute Java code

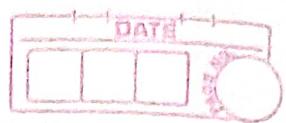
- It
- JVM is most imp part in java it convert our byte code to the native code
- JVM execute Java code from 3 components
 - ① class loader subsystem
 - ② runtime data areas
 - ③ Execution engine



- ① Class Loader subsystem
 - ① Bootstrap class loader
 - It load the class files from rt.jar file to JVM.
 - ② Extension class loader
 - It also load the class file but from ext folder in lib folder
 - ③ System class loader
 - It load our written class file to the JUM
 - ④ Custom class loader
 - From this we can load our own custom class.

- ② Runtime data area
 - ① Method Area
 - In this the class level data is stored.
 - ② Heap
 - Java object and their associated data are allocated.
 - ③ Java Stacks
 - local variable storage
 - ④ PC Registers
 - hold the address of currently executed
 - ⑤ Native method stacks
 - Method calls for C/C++

- ③ Execution engine
 - ① Interpreted
 - read and execute java bytecode one at a time.



② Just In time compiler

- If there is recursive function that is to do it recursively so for efficient used.

③ Garbage collector

- manage memory and delete memory that is not in use.

Q4 Explain the memory management System of the JVM?

⇒ In JVM memory management system is used to efficiently handle memory

① Heap

- All the Java object and arrays are allocated
It is the runtime environment

② Method area

- hold metadata about classes & methods
such as class definitions, method code & static variable

③ Java stacks

- Each threads has its own Java stack
so frames representing method calls and local variable

④ Program counter Register

- hold the address of the current instruction being executed by the thread.



③ Native method stack

- execute native methods with C/C++.

Q5) what are the JIT compiler and its role in the JVM? what is the bytecode and why is important for java?

Ans)

- JIT compiler is the compiler used in JVM for frequently executing the things like recursion.

- It is used for performance enhancement of execution engine.

- bytecode is the code come after compilation of java file and it store in .class file.

The JIT compiler convert bytecode to the native code.

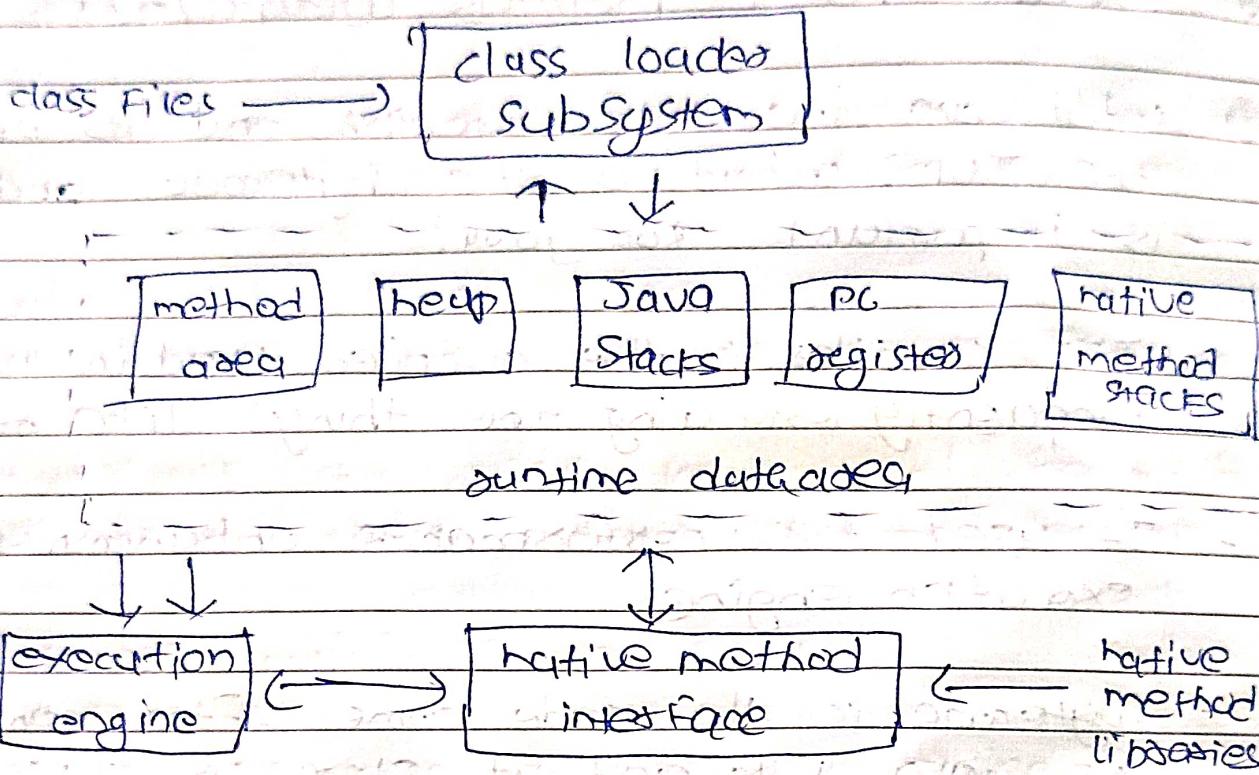
- byte code is not OS dependent but native code is OS dependent.

- byte code can run on any OS and it's binary file.



Q6) describe the architecture of the JVm?

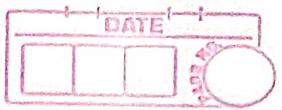
=>



- => There are 3 main component of JVm
- (1) class loader subsystem
 - (2) runtime data area
 - (3) native method interface

I have explained in Q3 The architecture

- Q7) How does Java achieve platform independence through the JVm?
- => Java is platform independent through JVm because IF we write a java program and compile it with javac.
- we will get .class file and this .class



File is platform independent.

- That .class file can run on any operating system just with the requirement that JRE is required.
- JRE is required then we can easily run our .class file i.e. byte code in any system.

Q8) What is the significance of the class loader in Java? what is the process of garbage collection in Java?

- class loader is the ~~most~~ component which is used to load the .class files to the JVM.
- Either the class files are loaded from ~~file~~ dt.jar or from exe folder in lib or our written class file.

class loader have 4 main components i.e

① Bootstrap class loader

- It loads the class file from dt.jar

② Extension class loader

- It loads the class file from ~~exe~~ folder

③ System class loader

- It loads the class file of System i.e. ~~java~~ ~~java~~

④ custom class loader

- It loads the class file with made by custom.

Garbage collector ~~not~~ remove the memory occupied which has no used by application.

- It prevent the memory leakage and optimize our memory.
- It use various algorithm to remove the data.
- one of that are parallel GC

Q) Q) what are the 4 access modifiers in Java and how do they differ from each other.

=> Access modifiers are used to control the visibility of the members of class(interface).

- 4 access modifiers in Java
 - ① private
 - ② package level private (also called default)
 - ③ protected
 - ④ public

Access modifier same class Sub class non sub class
non class

Private	Protected	Public	Default
P	P	P	P
P	P	A	A
P	A	A	A
P	A	A	A

Q) what is the difference b/w public protected & default access modifiers.

- =>
- ① public
 - public access modifier code accessed in all the classes, subclasses, packages.



- public means all can access it from anywhere

② protected

- protected is used in ~~same class~~ package same package Same class, subclass, non subclass i.e. in package anywhere but also in different package Sub class we can access it

③ private

- private means access in simple one class.

Q) 12) Can you override method with different access modifiers? Explain?

=> No we cannot override a method with a different more restrictive access modifier in a subclass.

• we can override a protected, ~~public~~ in subl method with protected or public methods in a subclass.

- we can't override a protected method with a private method because private is made less restrictive and not visible to other classes.

Q13 Is it possible to make a class private in Java? If yes where can it be done and what are the limitations?

- Yes it is possible but the class must be inner class
- If the top level class is private it will not work.
- So top level classes cannot be private; they can only be public or package private.
- Inner class can be private which restricts access to the class to the containing outer class only.

Q14 Can a top level class in Java be declared as protected or private? Why or why not?

- No top classes cannot be declared as protected or private.
- Top class can be declared as public or package level private they cannot be declared as protected or private because protected is not used for top class.



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

⇒

package private access is the access level that applies when no explicit access modifier is specified.

- It means access only in same package