

① Diffⁿ between compiler and interpreter

- Compiler :- compiler converts high level language into machine level language at once before the program runs.
- compiled code runs faster as compared to interpreted code
 - can improve security - CPU utilization is more
 - provides debugging tools, helps in fixing errors.
 - compiler runs faster, output: .exe
 - a small change in code needs to recompile whole file.
- Interpreter :- Interpreter converts HLL into MLL line by line before the program runs.
- does not require s.c. for later execution.
 - can convert only one sentence of program at a time
 - interpreted programs are easy to debug
 - interpreted language is more flexible than compiled language
 - runs slower
 - output :- no output.
 - a small change does not require translation of whole file
 - requires source code for later execution.
 - CPU utilization is less

② Difference between JDK, JRE and JVM

- JDK is java development kit for developing java applications, it contains JRE, ~~entry~~ loader, compiler and other tools required for developing a java application.

JRE stands for java runtime environments which provides an environment with minimum requirements to run java program. consist of jvm, core classes and binaries.

JVM is java virtual machine, it is responsible to run java code line by line

(3) How many types of memory areas are allocated by JVM
→ The memory in JVM into 5 diffⁿ parts

(1) class area / method area :-

- stores class code, variables (static), ^{function} method code, and class level data of every class such as runtime constant pool, field and method attributes.

(2) Heap :-

- objects created and stored
- used to allocate memory to objects at run time.

(3) Stack :-

- stores stack frame, each frame is created on calling a method thus also stores method data, the data is cleaned once method gets executed completely.

(4) Program Counter Register (PC Register) :-

every jvm thread that comes out a task of specific method has a pc register associated with it, and is capable of storing the return address or a native pointer on some specific platform.

(5) Native method stacks :-

- holds info about native methods of program
- also called as c stacks.

~~(6) Native method interface :-~~

(6) class loader :- when we run our java program the class file from harddisk gets loaded into memory.

Q4) What is JIT compiler :-

- - It is a component of the runtime that improves the performance of java application by compiling bytecode to native machine code at run time.
- If any is called again and again then interpreter need not to interpret it again and again, JIT compiler will take care of that methods thus improves efficiency.

Q5) access Specifier in java

→

- ① default :- can be accessed within same class & package.
- ② public :- can be accessed everywhere in application.
- ③ private :- can be accessed only within class.
- ④ protected :- can be accessed with same class, package, and outside package by subclass only.

Q6) What is compiler in java

- a compiler is a program that takes the text file works of a developer and compiles it into platform independent java file.

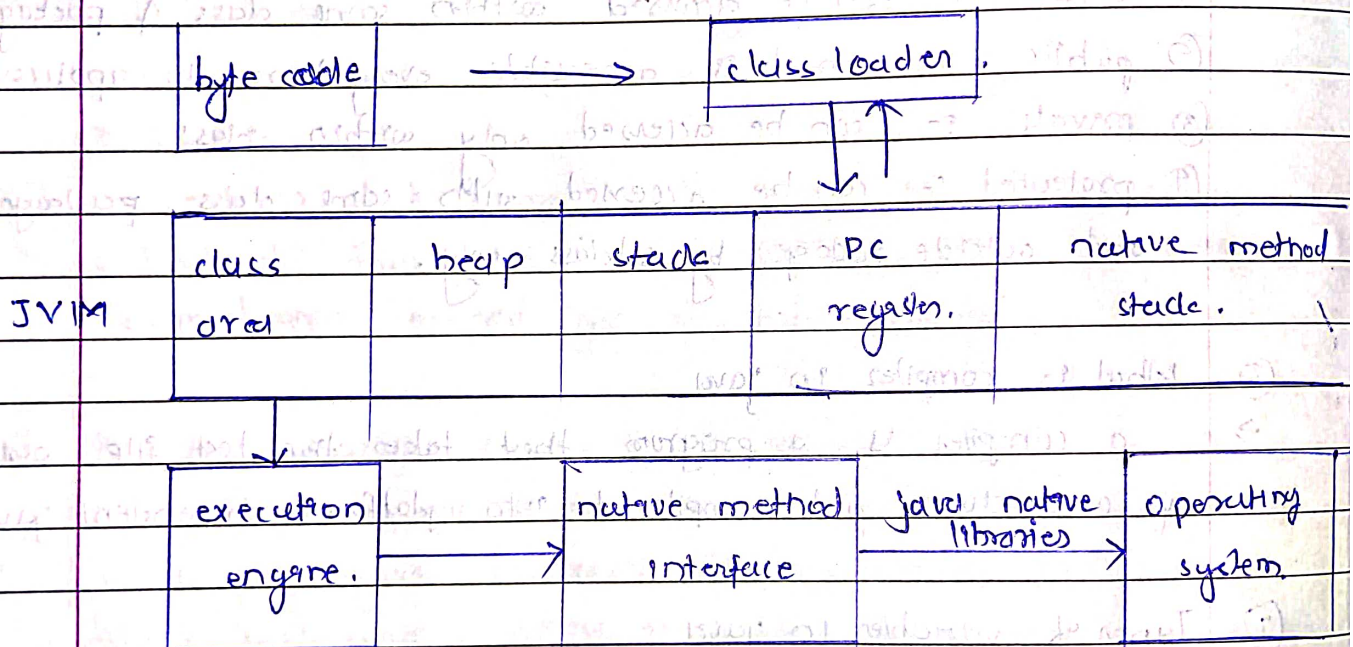
Q7) Types of variables in java

- ① local variables :- name given to a memory location, its value can be changed & re-defined within block or method.
- ② instance variables :- ^{non-static} class level, declared in class outside method body.
- ③ static variables :- declared similarly to instance variables, the diffⁿ is that static variables are declared using static keyword within a class outside of any method, we have one copy of static variable per class. irrespective of how many obj we create.

Q10

Architecture of JVM :-

- It is java virtual machine
- it is responsible for executing our java programs.
- It is a virtual layer present above the OS
- It performs four functions / operations :
 - a) loads code
 - b) verifies code
 - c) executes code
 - d) provides runtime environment.



after compilation of .java file we get its corresponding .class file

classloader :- when we run our java program .class file from the harddisk gets loaded in the memory. The loading task of .class file is performed by the class loader. after loading JVM allocates separate regions for our program.