Assignment No.: 2 PAGE NO.: DATE: / / 20 (1)

Que 1: Explain Component of JDK?

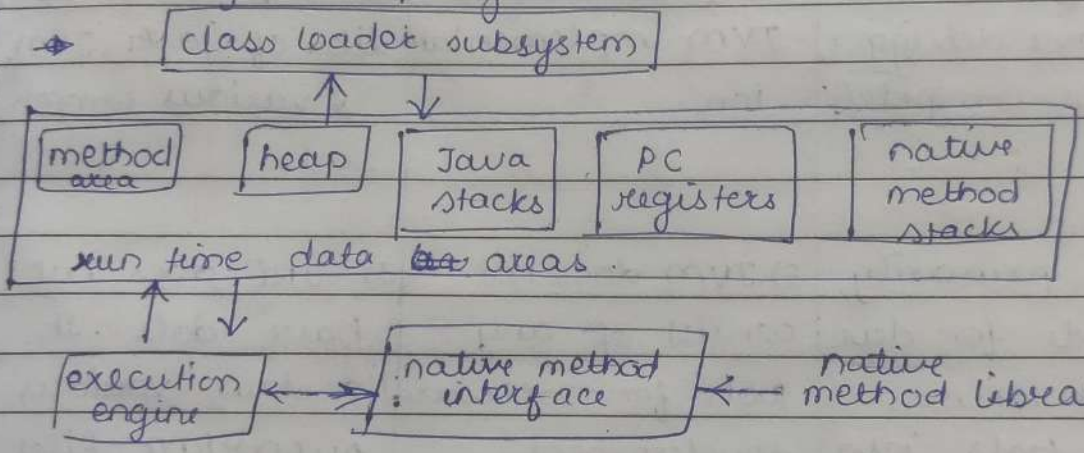
→ JDK is java development kit is used to develop java applications.

↳ Components of JDK are :-
jre, jvm, javac, api, debugger, javadoc,

1) JVM : Java virtual m/c.

→ Java applicatn run on any platform because of JVM

→ JVM interpreets bytecode into m/c code executed in java program.

→ 

- Class loader :→ is a subsystem of JVM.
  ↳ used to load class files.
  ↳ java program runs first loads class loader.

- Method area :→ Is data area in JVM in which class data will stored
  ↳ i.e static variable, Static blocks, static method, instance method.

- Heap :→ Is created when JVM starts up.
  ↳ It increase or decrease in size while applications runs.

DATE: / / 20

- Stack :→ JVM stack is also known as thread stack
  - ↳ JVM memory which created for single execution thread.
  - ↳ it stored local variables, partial results & data for calling method and returns.

- Native method stack :→ subsumes all native methods used in your application.

- Execution Engine :→ 1. JIT Compiler 2. Garbage collector.

1) JIT compiler :→ JIT is just in time compiler.
   - ↳ is a part of runtime environment.
   - ↳ improve performance of Java application by compiling bytecode to machine code at run time.

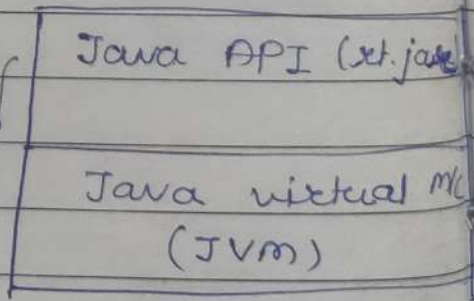
2) Garbage collector :→ collect unused material.
   - ↳ JVM done by garbage collector
   - ↳ tracks each & every object available in JVM removes unwanted ones

2) JRE : Java Runtime environment.

→ Can disk system takes java code combines it with needed libraries & start JVM to execute it.

→ JRE contains libraries & software needed by program to run.

→ JRE is part of JDK but download separately.



~~JDK is java based~~

3) Java development kit :(JDK)

↳ Is a software development environment used to develop java applications & applet.

↳ It contains JRE & several development tools ;

↳ interpreter / loader (java), compiler (javac), an archiver (jar), a document-ation generator (javadoc) accompanied with another tool.

Que 2:- Difference between JDK, JVM, and JRE?

JDK	JVM	JRE
1) JDK is java development kit.	1) JVM is java virtual machine	1) JRE is java run time environment
2) JDK is software development kit that develops applications in Java. It has various development tools: java debugger, java doc, compilers, etc.	2) It is a platform independent abstract machine. It describe the requirement of JVM implementation.	2) It is a implementation of JVM. It is a type of software package that provides class lib. of java, JVM & various component.
3) JDK is primarily responsible for development. It has various tools java, debugger, java doc, compiler etc.	3) JVM does not consist of any tools for software development.	3) JRE does not have tools. It contain various supporting files for JVM & class.
4) It is dependent means required different JDK for every diff platform.	4) It is independent means won't require diff. JVM for every platform	4) It is also dependent & require different JRE for every diff platform.
5) JDK primarily assist in exe. code.	5) Specifies all implementation. It responsible for providing all implementation to JRE.	5) Major responsibilities for creating environment for exe. code.

Que 3:
→ JV
→ JR
↳ JR
↳ w
wh
in
↳ A
or
↳
ja
(

PAGE NO.:
DATE: / / 20

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

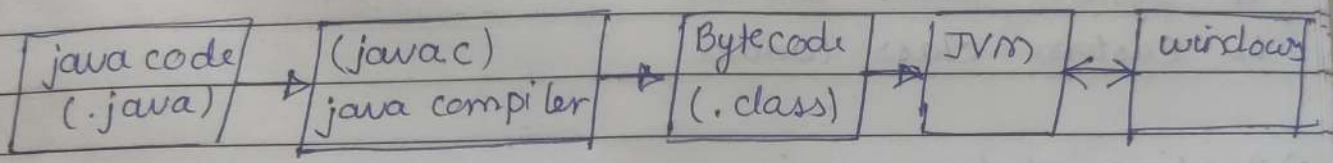
→ Role of JVM:-

↳ Java Interpreter is also called as JVM.

↳ where, java compiled into java bytecode which translated into special platform by interpreted java interpreter.

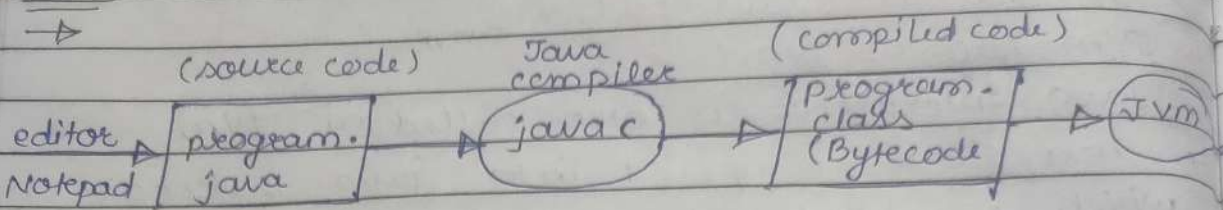
↳ Abstract machine designed to be implemented on top of existing processors.

↳ Hides underlying OS from java applications.



Que 4: Describe architecture of JVM:

→



• Component of JVM:

1) Class loader Subsystem
   - ↳ Bootstrap class loader
   - ↳ Extension class loader
   - ↳ System class loader
   - ↳ Custom class loader

2) Runtime data areas:-
   - ↳ Method area
   - ↳ Heap
   - ↳ Java Stacks
   - ↳ pc registers
   - ↳ Native method stacks.

3) Execution engine:-
   - ↳ Interpreter
   - ↳ Just-in-time (JIT) compiler
   - ↳ Garbage collector.

- javac → Java compiler which read java source code & convert into bytecode.
  - ↳ bytecode in .class file.
- .class file → created per type (interface/class/enum) defined inside .java file. ↳ source file & .class file name should not be same.

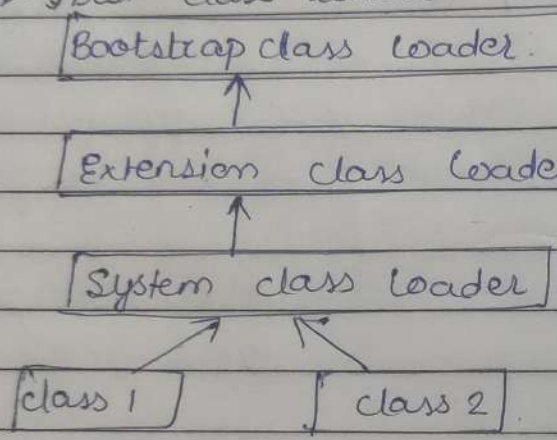
- if .java file is empty then java compiler do not generate .class file.

- Bytecode in java is not a m/c code rather it is obj. oriented assembly lang. code designed for JVM.

- if we want to disassemble the bytecode then we should javap tool.

ques: what is significance of class loader in Java? what is the process of garbage collection in java?

→ class loader is an abstract class.
↳ belong to java.lang package.
↳ loads classes from different resources.
↳ is used to load classes at run time.
↳ classes loaded into JVM according to need.
↳ First class loader run when java program exec



1) Bootstrap class loader :- Starts operation when JVM call
↳ it is not java class.
↳ Bootstrap class loader loads classes from rt.jar location.

2) Extension class loader :- Child of Bootstrap class load
& loads the extension of core java classes from JDK extension library.
↳ loads files from jre / lib / ext directory.

3) System class loaders :- is also called as system cl
↳ child of extension class loader.
↳ loads application type classes found in envir
-ent.

PAGE NO.:
DATE: / / 20

Garbage collection :-

→ It is automated process of deleting code thats no longer needed or used.

↳ free up memory space & ideally makes coding java apps easier for developers.

Garbage collection process use mark & sweep algorith

↳ there are two phases in algorithm : mark-n :- followed by sweep.

↳ Java obj created in heap, it has mark bit & i.e set to (0) false.

↳ mark phase :- garbage collector traverses object trees starting at their roots.

- when object reachable from root mark bit is set to (1) true.

↳ sweep phase :- garbage collector traverses heap reclaiming memory from all items with mark bit of (0) false.