Assignment -2

Q1 Difference Between JDK, JRE, JVM.

JDK	JRE	JVM
• "Java Development Kit"	• "Java Run Time Environment"	• "Java Virtual Machine"
• Often called "Super Set of JRE."	• Set of software tools responsiBle for execution of Java program (or) Application.	• It loads, Verify and execute Java Byte Code.
• It is foundational component, that enable "Java Application" and "Java Applet Development".	• Responsible for Creating an environment for execution of Java Code.	• It is an abstract (or) Virtual machine. • Also known as "Interpreter", Because it execute code line-by-line.
• Contains All Tools required to Compile, DeBug, run a program developed using Java plateform.	• Also called as "Java RTE". • Uses Various Package & their classes like (util, lang, math, io)	• Especially Responsible for Converting Byte Code to Machine specific code & it is necessary in both [JDK, JRE].
• It is "Software Development kit".	• It is an "Implementation of JVM".	
• Plateform Dependent; Because there are different JDK for different (O.S.).	• Uses heap space for dynamic memory allocation for Java objects.	• Plateform Dependent. • JIT (Just in Time) Compiler is Part of JVM
• JDK = JRE + Compiler + Development Tools	• JRE = Libraries + JVM + API	• JVM = Only need Run time Environment

Q2) What is JIT Compiler!

Ans) ① It is an "Integral Part of JVM". and stands for Just in Time compiler.

② It optimizes performance of Java application at compile or run time by compiling Byte code to ~~to~~ its native Machine code at run time.

③ It gets enaBled by Default.

④ When method is Compiled, JVM calls Compiled Code of that method directly, instead of interpreting it.

* Advantages :- ① Require less memory Usage
② Increase production efficiency, Competitiveness by minimizing (Waiting time, transport cost)
③ code Optimization done at run-time
④ Reduce page faults.

DiSadvantages :- ① Increase complexity of program.
② The program with less line of code does not take benefits of the JIT compiler.
③ It uses lot of Cache memory.

Q3) What is class loader?

Ans) 1.) Java classloader is an Abstract Class.

2) Part of JRE, it dynamically loads Java classes into JVM. at runtime.

3) Belongs to java.lang Package.

4) Java Classloader loads One Class at a time and also when it is required.

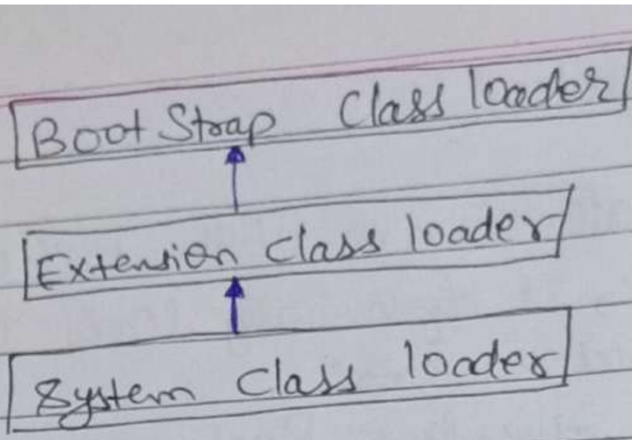
5) Classloader helps particular class to be loaded in memory; when needed by programmer for its implementation in code.

6) Java Classloader is based On (3) principle:

ⓐ Delegation:- It forwards the request for class loading to parent class loader.

ⓑ Visibility:- Allows child classloader to see all the classes loaded by Parent Classloader, But parent Classloader cannot see classes loaded by child class loader.

ⓒ Uniqueness:- Allows to load a class Once.
· It is achieved by delegation principle.
· It ensures that child class loader doesnot reload the class; which is already loaded by Parent class.



Hierarchy of class loader,

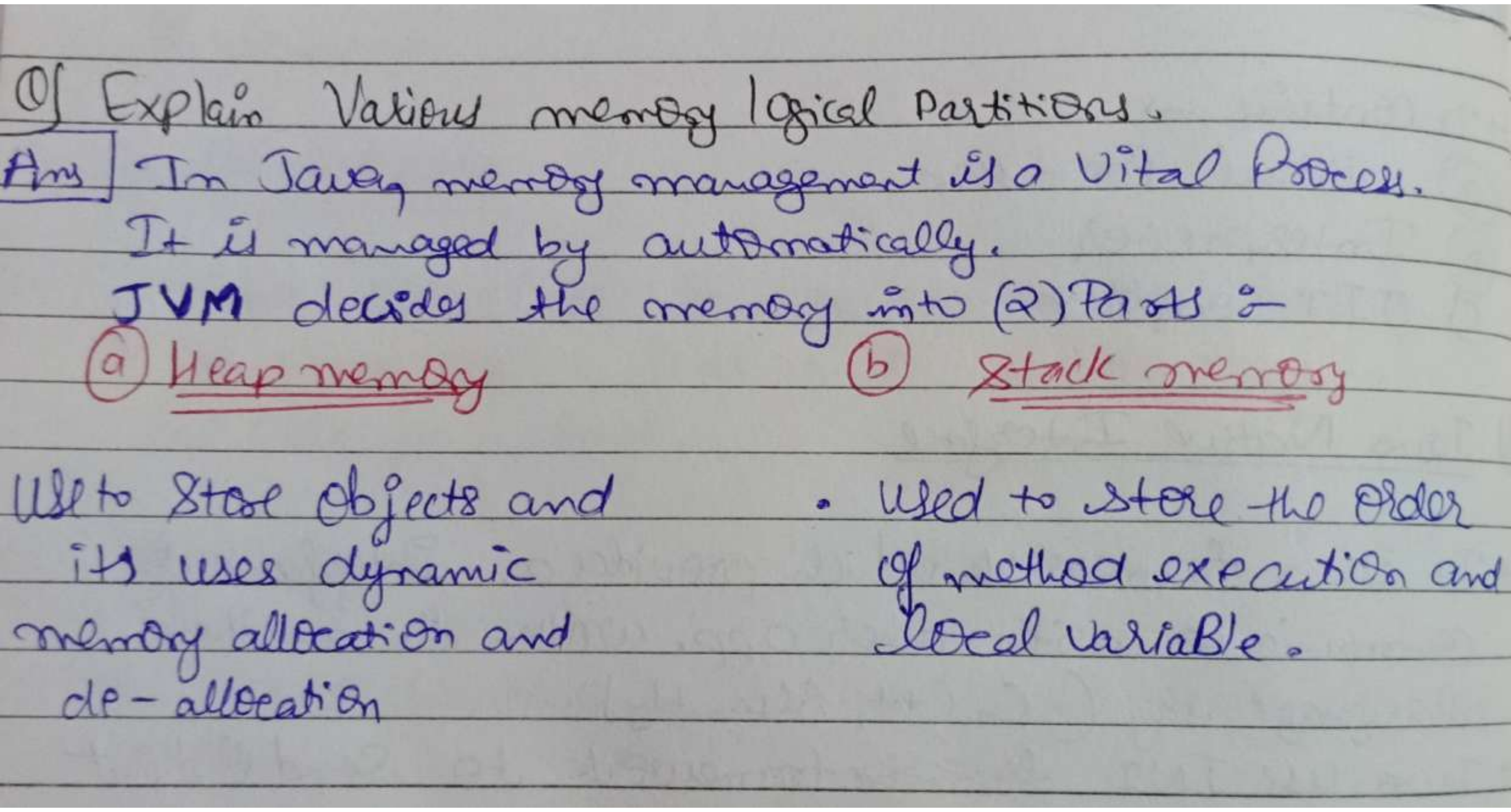
① BootStrap Class loader :- ① Also called Primordial classloader Because it Doesnot have any Parent class loader.

② It is Machine Code; which kickstart operation when JVM calls it.

③ Its job is to load the (1st) Pure Java Class loader.

② Extension Class loader :- ① It is child of BootStrap Classloader.
② loads extensions of core java classes from respative JDK extention library.
③ loads files from [jre/lib/ext] Directory

③ System Class Loader :- ① Also called Application Class loader.
② Child of Extension Class loader
③ loads application type classes found in enviroment variaBle CLASSPATH, -classpath, -cp command line Option.



Q] Explain Various memory lgical Partitions.

Ans] In Java, memory management is a Vital Process.
It is managed by automatically.
JVM decides the memory into (2) Parts :-

(a) Heap memory

Use to Store objects and its uses dynamic memory allocation and de-allocation

(b) Stack memory

- Used to store the order of method execution and local variable.

Stack memory

- JVM creates (Stack) memory for each thread.
- It is Physical Space (in RAM) allocated to each Thread at runtime.
- It is created when thread creates, memory management in the Stack follow LIFO order.
- Because, it is accessible globally; It stores variable references to object, and partial results.
- Memory allocated to Stack lives until function returns.
- If there is no space for creating new object, it throws an error java.lang.StackOverFlowError.
- Scope of element is limited to their threads.

Heap memory

- It is created when JVM StartsUp & used By Application as long as the application runs.
- It Stores object and JRE classes.
- whenever we create object of a class, it occupies space in the heap; while reference of that object gets memory in Stack.
- Doesnot follow any Order like Stack do.
- It dynamically handles the memory Blocks.
- for managing memory automatically, java provides Garbage Collector; that release memory of object which are no longer in use.
- Its memory lives, until any event, either Program terminates or memory free does not occur.
- It is common memory space shared with all the Threads.

Q4] What gives Java [WORA [write Once, Run anywhere] nature]?

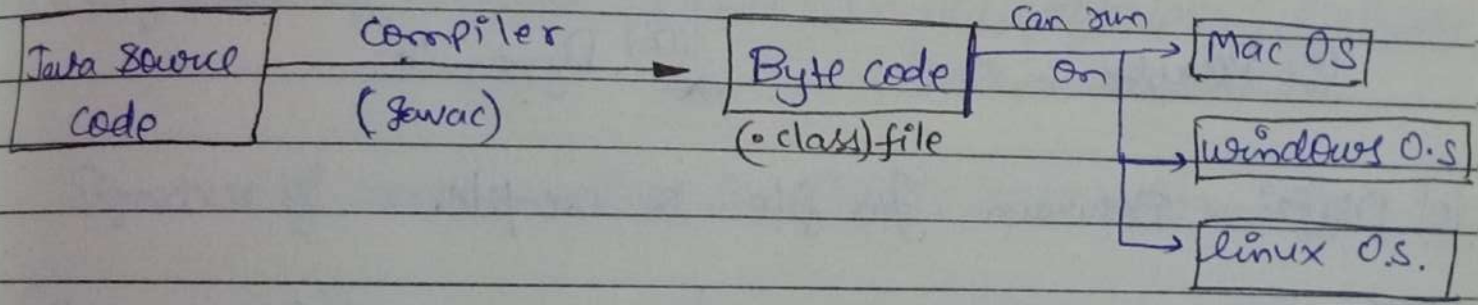
Ans] ① Java gets its WORA nature from ByteCode.

② Java Compiler convert source code to Byte Code.

③ Byte code is a type of intermediate level code; which is non-runnable.

④ Means, we can write Java code on any device; and then class file (Byte code) will be generated & this Byte code is able to run on any plateform.

⑤ Bytecode also give Java another feature; of being Plateform independent.



⑥ As Bytecode gets genekated; JVM comes in pitture; JVM is responsible for interpreting Byte code to machine understable code; & JVM comes with JDK; thus JDK is plateform Dependent [means; Different JDK for Different O.S. plateform].

Q6 "History of Java".

Ans) → Java was Created By "SunMicroSystems".

→ James Gosling led a team of Researchers in an effort to create a new language; that allow consumer electronic device to communicate with Other.

→ work on language began in (1991).

→ Java Team members also called (Green Team).

→ Basically made for "electronic Device Communication".

→ Principles for Creating Java ⇒ Simple ; RoBust ; Portable; platform independent; Secured ; High Performance; MultiThreaded; Architecture Nutral, Object-Oriented ; Interpreted; Dynamic.

Q7) What is Original Name of Java? Why it is Renamed?

→ Initially; (Java) was Named as (Green Talk) → (1st) name ; (.gt) extension

→ later ; it called [Oak].

⇒ Why Oak? ⇒ (Oak) is [Symbol of Strength] and chosen as a national tree of many countries [USA, France, Germany].

→ later, named [Java] ; Bcoz ; (Oak) name was Trademark By "Oak Technologies."

⇒ Why Chosen Java? ⇒ Java is an Island in Indonesia; where (1st) coffee was produced called Java coffee.

- It is kind of espresso Bean.

Q8 List features of Java / OOP ?

Ans
1. Simple
2. Object Oriented
3. PortaBle
4. Plateform independent
5. Secured
6. Robust
7. Architectural Neutral
8. Interpreted
9. High Performance
10. Multi Threaded
11. DistriButed
12. Dynamic

1) Simple :- (Simple, clean, clear) Syntax. It is Because

① Java Syntax is Based On C++.

② Java removed complicated & rareUsed features (Pointers, Operator overloading)

③ No Need of removing Unreferenced Objects Because there is a (Automatic Garbage Collection in Java).

2) Object-Oriented :- Means We can Organize Our Software as a Combination of different types of Objects that incorporate Both (Data & Behaviour).

* Basic Concepts of OOPs :-

① Object ② Class ③ Encapsulation

④ Polymorphism ⑤ Abstraction ⑥ Inheritance

③ Platform Independent :- Java is WORA; Java Provides Software Based-Platform

It has (2) Components :-

① Runtime Environment

② API (Application Programming Interface)

-> Java Code Can executed on Any Platform [windows, Mac, Linux, Sun Solaris].

-> Source code of Java is Compiled & Converted to Byte Code.

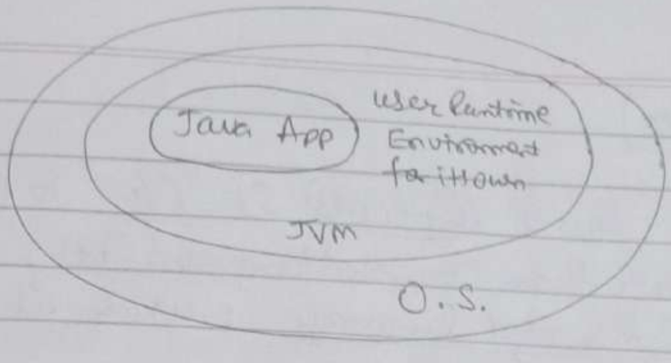
-> This Byte code is Platform Independent [means Can Be executed On any Platform.]

④ Secured

- Java is Well Known for its Security; use to Develop "Virus-free System"

It is Because; ① No explicit Pointer

② Java Program run inside Virtual Machine SandBox



③ Class loaders :- Part of (JRE); used to load Classes of Java into (JVM) dynamically.

Imp { It Adds Security By Separating package for the Classes of local file system from those that are imported from Network sources. }.

④ ByteCode Verifier :- Checks the Code fragments for illegal Code that violate access rights to Objects.

⑤ Security Manager :- Determines what resources a class Can access such as (Reading, writing) to local Disk.

⑤ Robust

- Uses Strong Memory Management
- There is lack of pointers that avoids Security Problem.
- Provides Automatic GarBage Collector; that runs On JVM; to get rid of objects which are Not Being Used By Java App.
- There are exception Handling & type checking mechanism.

⑥ Architectural-Neutral

- There is NO implementation Dependent features; example Size of primitive types are fixed.

eg:- In Java; (int) occupy (4 Bytes) for Both (32, 64) Bit Architecture
in (C); (int) occupy (2) Byte for (32 Bit Architecture) &
(4) Byte for (64 Bit Architecture).

⑦ Portable → facilitate; you to carry Java Bytecode to any Platform.
Doesnot require any implementation.

(8) High Performance

- It is Because; Java's ByteCode is "Close" to (Native Code).
- Still little slower than Compiled language ((C++);
- Java is an Interpreted language ; which is Slower than Compiled language (C, C++).

9) DistriButed

- It facilitate Users to Create DistriButed Apps in Java.
- RMI, EJB are used for Creating DistriButed Apps.
- It make us able to access files By Calling the methods from any machine On the Internet.

(10) Dynamic

- It is Dynamic language; Support Dynamic loading of Classes
- Means; Classes Can Be loaded On Demand.
- Also support functions from its Native language (C, C++).

(11) MultiThreaded

→ Thread is like a Separate Program ; executing Concurrently.

→ Main Advantage of it is; It Doesnot Occupy memory for each Thread.

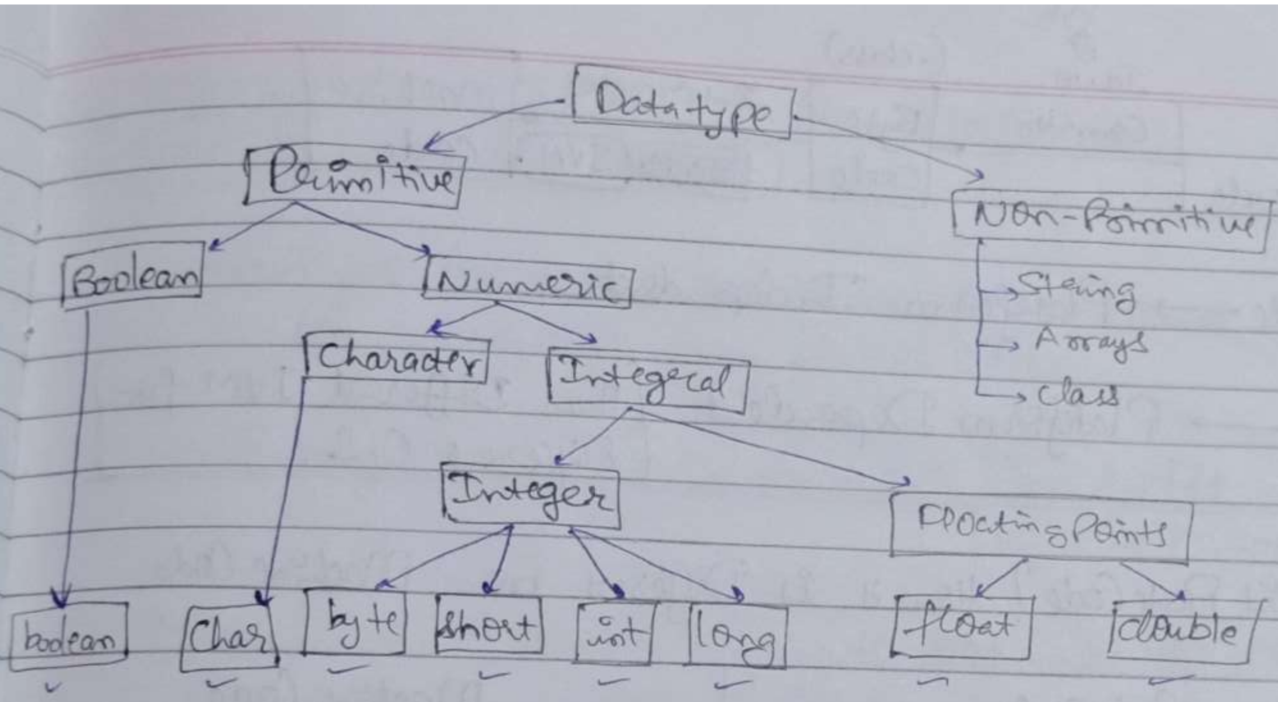
→ Shares Common Memory Area.

→ Important for Multi-Media ; WeB Apps.

Q9 list Various DataTypes in Java.

Ans ① Primitive :- boolean, Char, int, short, byte, long, float, double

② Non-Primitive :- class, interface, Arrays



	Default Size	Range	Default Values	Literals
boolean	(1) Bit	–	false	true or false
char	(2) Bytes	\u0000 to \uffff	'\u0000'	[illegible]
byte	(1) Bytes	-128 to 127	0	[illegible] ×
short	(2) Bytes	(-2^{15}) to $(2^{15}-1)$	0	×
int	(4) Bytes	(-2^{31}) to $(2^{31}-1)$	0	3, 007, 0xBAAC
long	(8) Bytes	(-2^{63}) to $(2^{63}-1)$	0L	3L
float	(4) Bytes	$(-\dots e^{45})$ to $(+\dots e^{38}-1)$	0.0f	3.0f, 3.0E2f
double	(8) Bytes	$(-1.7\times e^{324})$ to $(+1.7\times e^{308}-1)$	0.0d	3.0, 3.0E2

Q16) Difference B/w (System. out. print), (System. out. println)
(System. err. print).

Ans) System. out. print() retains cursor in the Same line after printing argument.

System. Out. println() moves cursor to Next line after printing argument.

System. err. print() it normally Used TO Print Error text.
• It is also a Print Stream.

Q10] How Java is Plateform Independent?

Ans] → Meaning of Plateform-Independent; means Java Compiled code (Byte code) can run on all O.S.

→ Steps ① Program Written in JAVA; (Javac) Compiles it

② Result of JAVA Compiler; it makes (.class) file @ ByteCode

and (Not) machine Native Code (unlike C compiler)

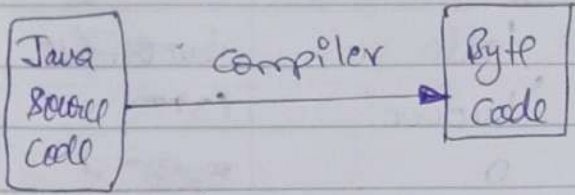
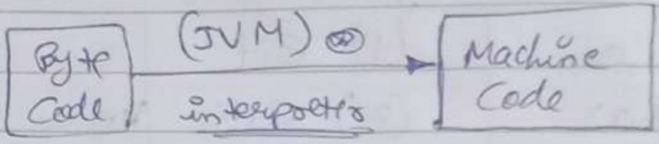
③ Bytecode generated is (Non-executaBle Code); Here needs an interpreter to execute on Machine. & this Interpreter is JVM!

Thus we need JVM to execute ByteCode.

④ finally gives desired Output.

Q11 What is Byte Code? How it is Different from Machine Code.

Ans

Byte Code	Machine Code
	
⇒ Contains (binary, Hexadecimal, macros (new, swap, add)); Not directly understable By CPU.	→ Contains (binary) instructions; directly understable By CPU
⇒ To read & execute this code; It require Interpreter.	→ (Machine code) considered as (low-level code) ⓞ Machine understable code ⓞ Machine level code.
⇒ Kind of Intermediate Code.	
→ Non-runnable code; generated by after compilation of source code.	⇒ It is set of instructions in Machine language (Binary format), directly executed by CPU.
→ Executed By (JVM), ~~then~~ (CPU).	⇒ Executed By CPU, not By JVM
→ less Specific towards Machine	→ More specific towards Machine
→ Plateform Independent;	→ Plateform Dependent.
⇒ All source code need not to Be converted to Byte code for CPU execution. source code written By specific High-level language converted to Byte code then to object code for execution By CPU.	→ Source code must Be converted into Machine code; Before executed By CPU. →

Q12) Differena b/w (Jar file) & (Runnable Jar file)

Ans) JAR file → Java Archieve

=> A file format based on Popular ZIP file format and it used for aggregating many files into One.

=> Primary Reason for its Development is {Java Applets and their requisite Components}.

→ Contain all of Various Components that make UP a Self-contained, executable Java App, deployable Java Applet; a Java library to which any JVM can link.

Benefits Using JAR file :- (1) Ability to Aggregate hundreds of different files ;to make an Application.

(2) Compress all of Contained files, greatly reducing size of Application; make it easier to Move JAR file Over Network & B/w environment.

→ JAR file is a Java App which require Command line to run.

=> Runnable JAR file

- This JAR file Can directly executed by DouBle Clicking it.

- Allow a Use to run Java classes without Having to know class names & type them in Command prompt.

- It allow Java Classes to Be loaded ; Just like when a User clicks on (exe) files.

Q) Difference Between Jar file & exe file.

Ans

Runnable Jar file	exe file
① Jar file are like Dead Body	① exe files are like living Body
2) It is Combination of compiled Java classes	2) Executable files are combination of compiled Java classes with main Class
③ JAR files are executed Only in (JRE).	③ exe files can be executed in (O.S. Environment).

Q14] How (C) is Plateform Dependant language ?

Ans] → C compiler is designed to Produce <u>Plateform - Specific</u>, <u>Optimized Code</u>

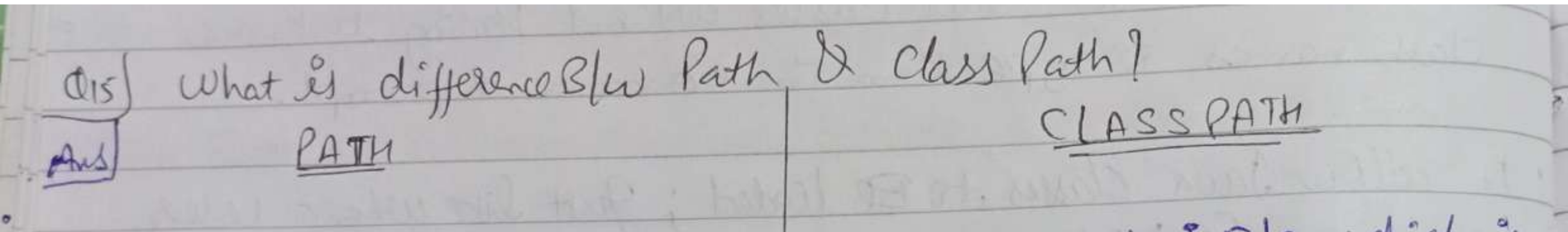
→ (C) language also called as (System language); Because it directly convert <u>source code</u> to <u>machine code</u>.

→ (C) is designed for Producing [fastest Performing code].

→ In (C); (<u>Machine Code</u>) is <u>different</u> for {different Processor architecture.}

⇒ Windows (C) compiler will only work for windows (O.S) and

→ MAC compiler will only work for MAC (O.S); thus make it Plateform Dependant.



Q15) What is difference B/w Path & Class Path?

Ans)

PATH	CLASSPATH
• Enviroment VariaBle, used By our O.S. to locate different files (.exe), (Java liBraries like (Javac, Java) command,	• Enviroment VariaBle, which is used by Java Compiler; to find path of classes. ie; when we talk about (J2EE) Java Enterprise Edition; we define JAR files Path.
• (path) is like we are setting	

enviroment for O.S., to look in this path for executaBles.

- Set of tools of Java in Java Program
- like (java, javac); to compile Program.

- To set PATH, we need to include JDK_HOME/bin directory in PATH enviroment variable.

- Once set; cannot Be overriden By Java setting

- Used By System/App classloader to locate & load compile Java Bytecodes stored in (.class) file.

- To set CLASSPATH; we need to include all those directories where you have put either (.class) file @ JAR File; required By Java Appl.

- Classpath values can Be overriden using Command-line option @ -cp @ -classpath to both javac, java Command.

- Java Compiler & JVM use CLASSPATH to determine location of Required class files.