Assignment 2 Jai Tembhane KH

Q1. What is diffrence between JDK, JRE and JVM?

Ans:-

Parameters	JDK	JRE	JVM
full-form	Java Development kit	Java Runtime Environment	Java Virtual Machine.
Defination	JDK is a software development kit that develops application in Java. Along with JRE, the JDK also consists of various development tools ex. JavaDoc Java Debugger.	JRE is an implimentation of JVM. It Provides class Liblaries of Java, JVM and various other components for running the applications written in Java programming.	JVM is an platform independent abstract machine that has three notations in the form of specifications. This document describes the requirement of JVM Implementation
Functionality	The JDK primerly assists in executing code. It primerly functions in development	JRE has a major responsiblity for creating an environment for the execution of code.	JVM specifies all of the implementations. It is responsible implementation to the JRE.
platform Dependency	Dependent	dependent	Indipendent
Implementation	JDK = Development Tools + JRE	JRE = Libraries for running the application + JVM	JVM = Only the runtime environment that helps in executing the Java bitcode.

Q2. What is JIT compiler?

Ans:- Just in Time :- The JIT compiler is a component of runtime environment that improves the performance of Java applications by Compiling bytecodes to native machinecode at runtime.

- Java program consists of classes, which contain platform-neutral bytecode that can be interpreted by a JVM on many diffrent computer Architectures. At run time, the JVM loads the class files, determines the semantics of each individual bytecode and perform the appropriate computation; The additional processor and memory usage during interpretation means that a Java application performs more slowly than a native application. The JIT Compiler helps improve the performance of Java Programs by Compiling bytecode into native machine code at run time.
- JIT compiler is enabled by default. When a method has been Compiled.
- JIT compilation does require processor time and memory usage.

$$\begin{tikzcd}
\text{Sourcecode.java} \arrow[r] & \text{Compiler} \arrow[r, "\text{At compile time}"] & \text{Bytecode} \arrow[d, "\text{At Run time}"] \\
& \text{Native Machine Code} & \text{JIT Compiler} \arrow[l]
\end{tikzcd}$$

Q3 What is class Loader?
Ans:- The Java class Loader is a part of Java Runtime Environment ~~that~~ that dynamically Loads Java classes into the Java virtual Machine. The Java runtime system does not need to know about files and file system because of classloader. Java classes aren't Loaded into memory all at once, but when required by an application. At this point, the Java classLoader is called by the JRE and these class Loader Load classes into memory dynamically.

Q4. Explain various memory logical partitions?
Ans:- memory partitioning means dividing the main memory into chunks of the same or different sizes so that they can be assigned to processes in the main memory
There are two types of memory partitioning techniques.
i) Fixed ~~or fixed memory~~ static memory partitioning.
2) Variable or Dynamic memory partitioning.
1) Fixed or Static memory partitioning!- the main memory is divided into blocks of the same or different sizes. Fixed memory partitioning can take place before executing any process or during the configuration of the system.
2) Variable or Dynamic partitionin!- Dynamic partitioning tries to overcome the problem caused by fixed partitioning. In this techinique, the partition size is not declared initially. It is decleared at the time of process loading.

Q5 What gives Java its 'write once and run anywhere' nature?
Ans:- The Bytecode. ~~Java~~ gives Java its write once + run anywhere nature.
Java compiler converts the Java programs into the class file (Byte code) which is the intermidiate Language between source code and machine code. The bytecode is not platform specific and can be executed on any computer.

Q6 Explain History of Java? who invented Java?
Ans. • Java was developed by James Gosling, who is known as the father of Java. in 1995
• James Gosling and his team members started the project in early 90's
• Currently Java is used in internet programming mobile device, games, e-business etc.
- James Gosling, Mike sheridan and Patric Naughton initiated the Java Language project in Jun 1991 The small team of Sun engineers called Green team.

- It was designed for small embaded Systems in electronic applinces like set-top boxes.
- Firstly it was called Greentalk by Gems Gosling & file extension was .gt
- After that, it was called Oak symbole of strength.
- In 1995. ~~Od~~ Oak was Renamed as Java.

Q7. What was orignal name of Java? why it was renamed?

Ans:- The language ~~called~~ was initally called Oak after an Oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffie, a type of ~~Go~~ coffie from Indonesia.

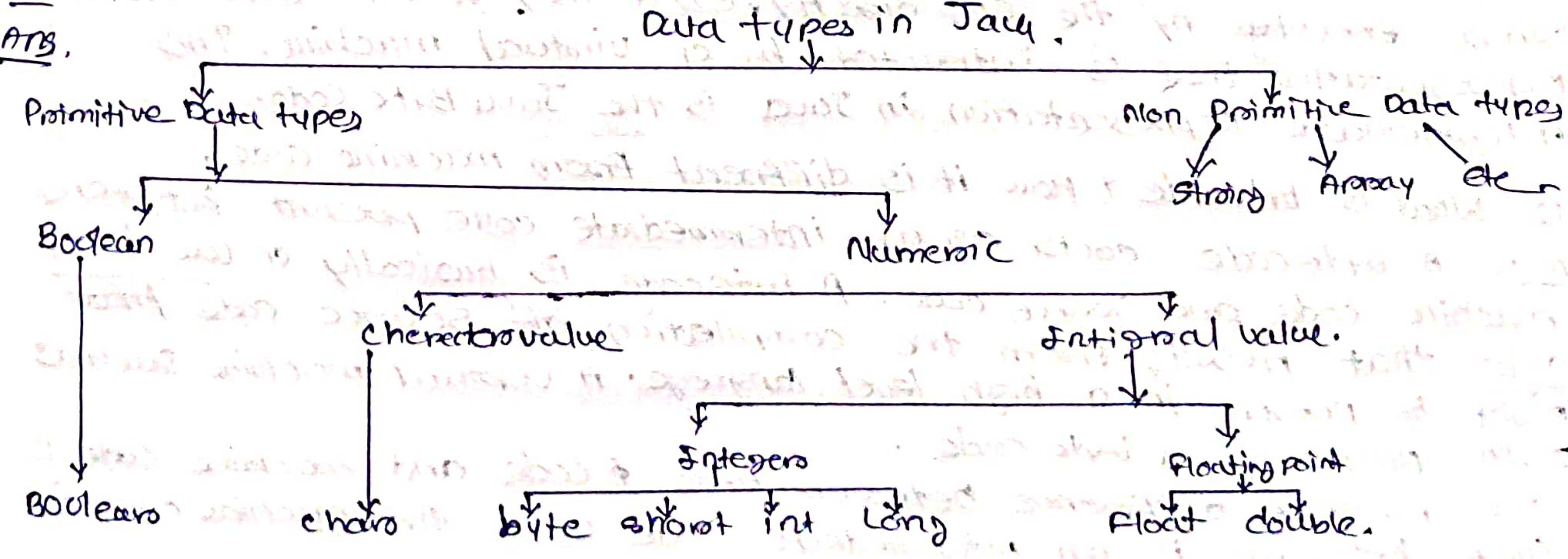
- ~~of~~ Oak is a symbol of strength and chosen as a national tree of many ~~cou~~ countries like the U.S.A., France, Germany, Romania, etc. In 1995. Oak was renamed as Java because it was already a treademark by Oak Technologies.

Q8. List features of Java?

Ans:- 1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Seccured
6. Robust
7. Architecture nature
8. Interpreted
9. High Performance
10. Multithreade
11. Distributed
12. Dynamic.

Q9. List various Datatypes in Java?

Ans.



Q10. What is diffrence between System.out.print, System.out.println, System.err.print?

System.out.print	System.out.println	System.err.print
will print standard output on the system	will print standard output & move courcers to next Line on the system.	will print to the standard errors
mostly used to disply result on the Console	mostly used to disply result on the console	mostly used to output errors texts..
it gives output on the Console with default (black) colour	it gives output on the console with default (black) colour	It also gives output on the console but most of the IDE's give it a red colour to diffrentiate

Q11. How is Java platform independent

Ans. The meaning of platform-independent is that the java compiled code (byte code) can run on all operating systems.
A program written in a Language that is a human-redable language. It may contain words, phrases, etc. which the machine does not understand. For the source code to be understood by the machine, it needs to be in a language understood by machines, typically a machine-level language. So heare comes the role of compilers. The compilers converts the high level language (human Language) into a format understood by the machines. Therefore, a compiler is a program that translates the source code for another program from a programming language into executable code.
This executable code may be a sequance of machine instructions that can be executed by the CPU directly, or it may be an intermediate representation that is intrupted by a virtual machine. This intermediate representation in Java is the Java Byte Code.

Q12 What is bytecode? How it is diffrent from machine code.

Ans:- A bytecode acts as an intermediate code present betweena machine code and source code. A bytecode is basically a low. level code that results from the compilation of source code that might be present in a high level language. A vertual machine such as JVM processes a byte code.

- The primery diffrence between byte & code and machine code is that bytecode is an intermediate code while the machine code is the final code that the CPU processes.
- machine code is present in a binary format of 0's and 1's thus complet diffrent from bytecode

Q13 What is diffrence between Jar file + Runnable jar file?

Ans:- A JAR ~~es~~ file is a Package file format typically used to aggregate many Java class files and associated metadata and resources into one file which requires comand line to run. ~~[illegible]~~.
While, A runnable jar file allows a use to run Java classes without having to know class names and type them in command prompt, rather than the users can just double click on the jar file & program will fire up. A runnable Jar allows Java classes to be loaded just like when a user clicks an exe file.

Q14 How is c platform dependent language.

Ans:- In c machine code is diffrent for diffrent processor architecture and thus could not run natively on incompatible platforms unless you have a emulator to help you out.

Q15. What is diffrence between path + Classpath.?

Ans:- Path is an environment variable, path that behaves as a mediators between the Operating System and developers to inform binary file path.
On the other hand ClassPath, is a Parameter in the JVM that is used by a system or application Classloaders to locate and Load compile Java bytecode stored in the ".Class" file.