Assignment - 2

1. Differences between JDK, JRE, JVM.

JDK, JRE and JVM are the three components of the Java programming language.

- JDK (Java development Kit) - It is a software developement platform that contains tools & libraries needed to create, compile & run Java applications. It includes JRE, debugger, compiler, etc. JDK is platform dependent i.e. it has different versions for different platforms.

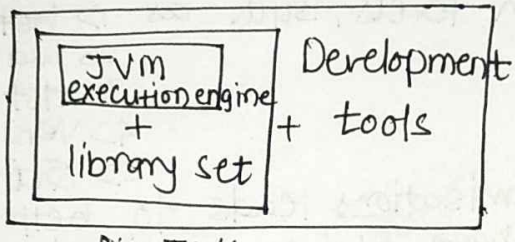


fig. JDK

- JRE (Java Runtime Environment) -: It is a software package that provides environment to only run, java programs on your machine. It includes JVM & other core libraries that support Java features. JRE is also platform dependent.

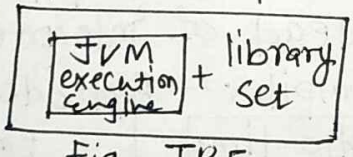


Fig. JRE

- JVM (Java Virtual machine) - It is a software component that executes the Java bytecode generated by the compiler. It acts as layer between Java code and underlying hardware & software. It ensures Java programs are platform independent.

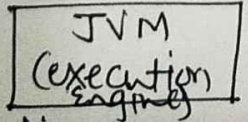


fig. JVM.

- loading of code
- verify
- execute the code

Q2) JIT compiler.

~~A 'Java In Time'~~

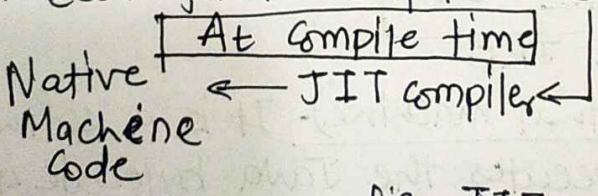
1) The 'Just In Time' compiler is an integral part of JRE. It is a program that improves the performance of a code(Java application) by compiling bytecodes to native machine code at run time.
Byte codes are platform-neutral instructions that can be interpreted by JVM. A JIT compiler converts bytecodes to instructions that are directly sent to CPU.

2) A JIT compiler compiles a method at different optimisation levels, such as 1) ~~Hot~~ Cold
2) Warm
3) Hot
4) Very hot
5) Scorching.

Higher optimisations leads to better performance but they have higher compilation cost in terms of CPU & Memory. A ~~JIT~~ JIT compiler uses heuristics to decide which methods to compile and at what level.

3) It is enabled bydefault in Java. When a method is called, JVM calls the compiled code of that method directly instead of interpreting it.

Sourcecode.java → Compiler → Bytcode



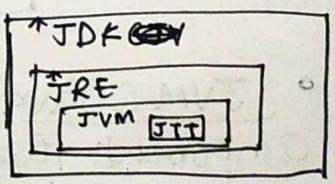


fig. JIT.

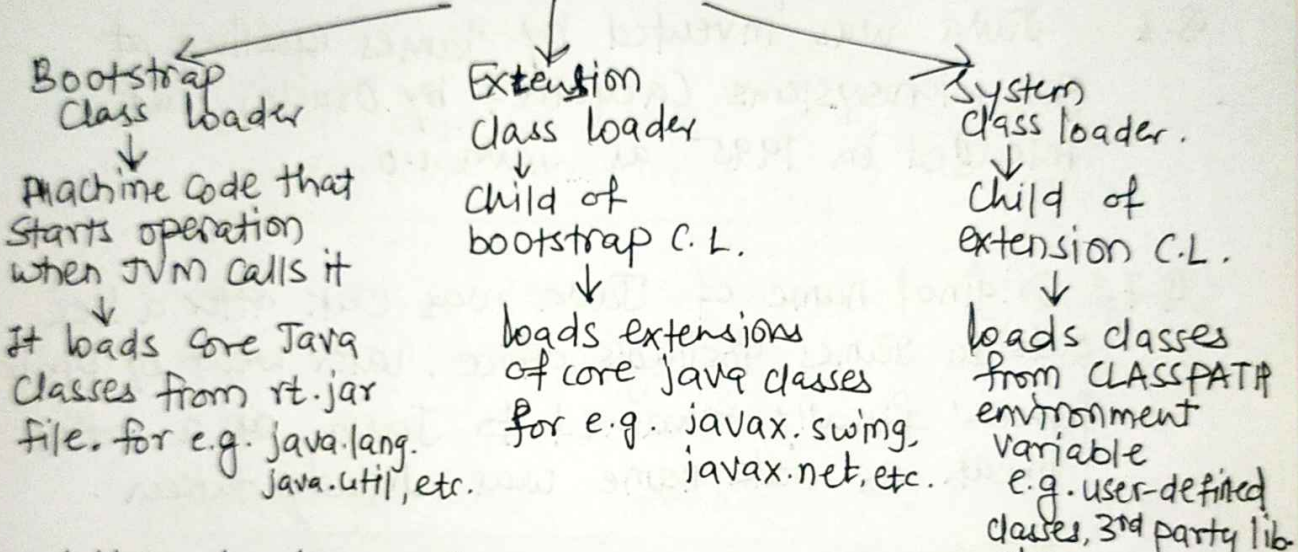
q.3. What is class loader in Java.

A Java classloader is a class that loads class files. Class files contains the compiled java code that contains bytcodes. A class loader dynamically loads Java class into JVM on demand, i.e. it need not know about files & file systems.

3 types of class loaders -:

~~1> Bootstrap classloader - It is a machine~~

Class loaders



When loader receives request to load, it follows 3 principles -> 1> Delegation

2> Visibility

3> Uniqueness

Q.-5

Bytecode gives Java WORA nature. Java ~~programs~~ compiler converts java programs to class file (Byte Code) which is intermediate language between source code & machine code. This bytecode is not platform specific and can be executed on any computer.

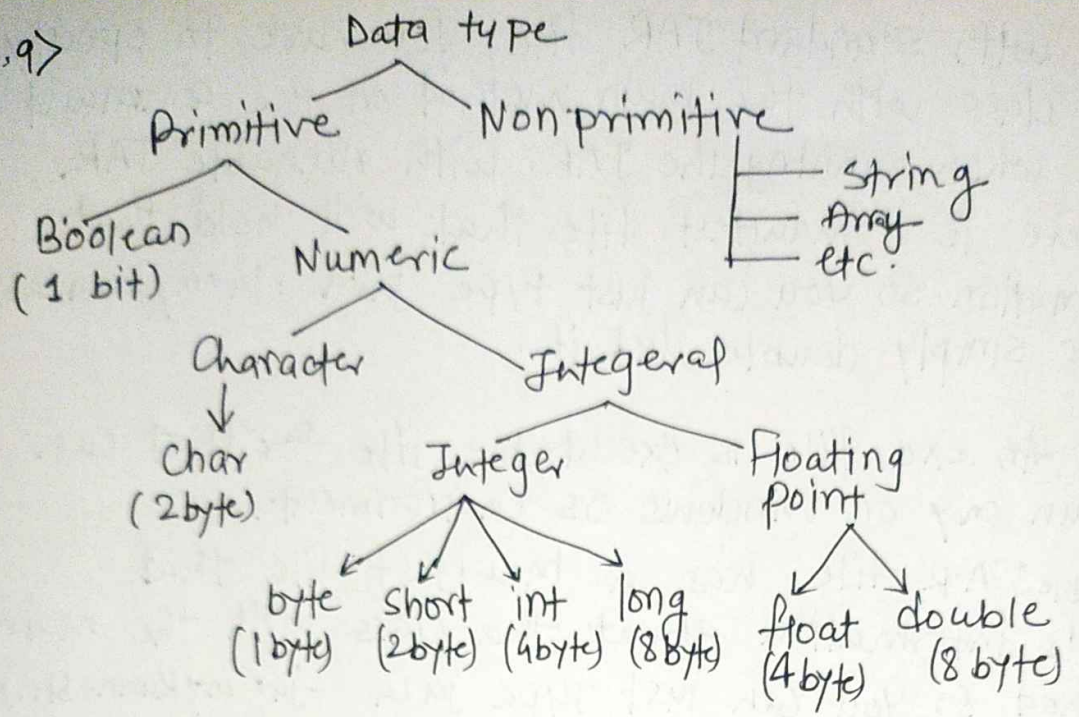
Q.6. Java was invented by James Gosling at Sun microsystems (acquired by Oracle) and released in 1995 as Java 1.0.

Q.7. Original name of Java was Oak after a tree outside James Gosling's office. Later went by project 'Green' finally renamed to Java as a coffee break as oak name was already taken.

Q.8. features of Java.

1> Simple
2> Object oriented - Modularity
3> Portable
4> Platform independent
5> Secure
6> Robust
7> Multithreaded
8> Dynamic
9> Distributed
10> ~~Arritech~~ Architechure neture (WORA)

Q.9>



Q.10> System.out.print - Standard Output Statement
System.out.println - Standard output Statement on newline

System.err.print - Standard error of system. (It does not print newline after the content)

Q.11> JVM acts as runtime engine that calls the main method present in the Java code. The .class file contains bytecode which is machine independent and it is compiled by JVM. It is close to native codes & hence Java is platform independent.

Q.12. - Bytecode is the instruction set for JVM.
- It acts similar to assembler in C++.
- When java program is compiled bytecode is generated in the form of .class file.
- Machine code is directly executed by CPU while bytecode is created after compiling source code and executed by the virtual machine.

Q.13. With standard JAR file, you have to specify the class with the main method on the command line when running the JAR. With runnable JAR, there is a manifest file that will hold that information so you can just type java -jar myRunnable.jar or simply double click it.

Q.14. An .exe file is executable file ~~for~~ that can run only on windows OS environment.

RunnableJAR file has a manifest file that holds information about the class with the main method so you can just type java -jar myRunnable.jar ~~or~~ simply double click it.

Q.15 Applications that are developed by using C cannot be executed on other OS, whenever we compile source code of program using C, it will generate machine code for specific hardware, software.

Q.16. In Java, PATH is an ~~envir~~ environment variable used to locate JDK binaries like java or javac commands used to run & compile java source file.

while CLASSPATH is an environment variable used by system to application classloader to locate & load compiled Java bytecodes in .class files.