JAVA ASSIGNMENT I

SHIVA KUMAR   20210004

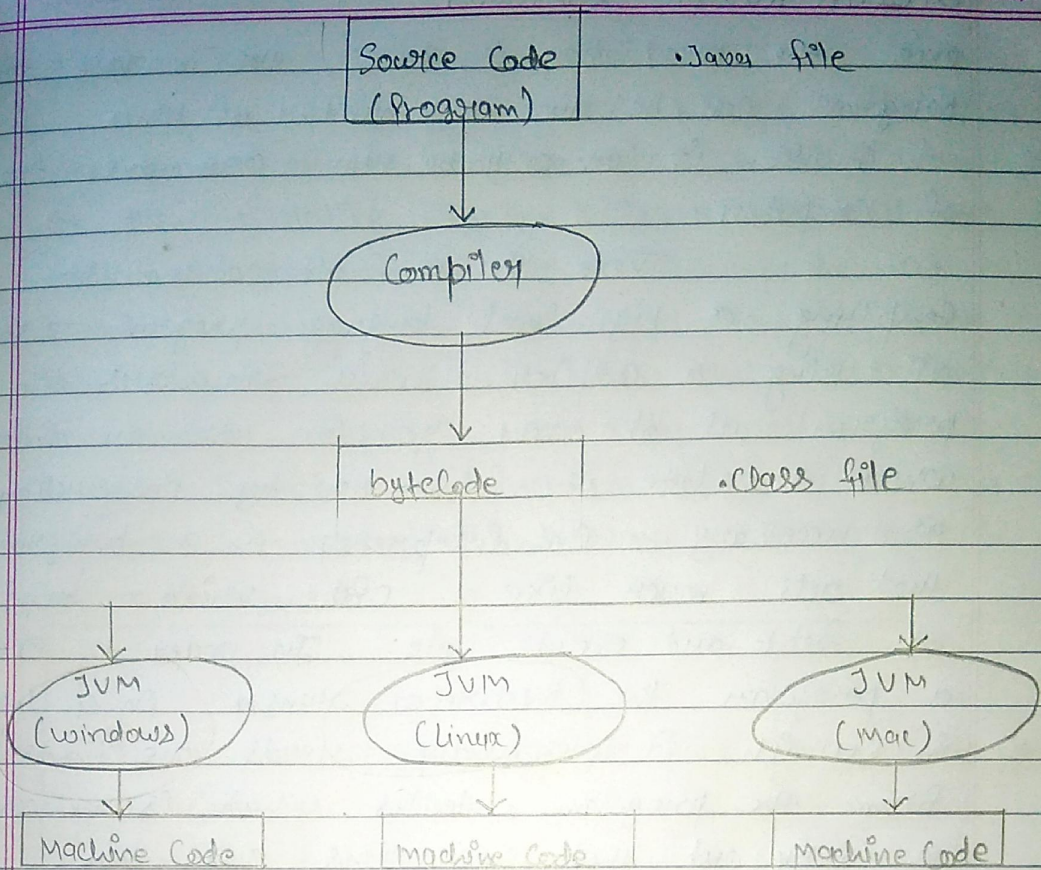1. Q   write a short note on:

1) Bytecode
2) JVM
3) JDK

1) Bytecode

Java bytecode is the instruction set of Java virtual machine. It acts similar to an assembler which is an alias representation of C++ code. As soon as a Java program is compiled, Java bytecode is generated. Java bytecode is the machine code in the form of a .Class file. With the help of Java bytecode we achieve platform independence in Java.

→ How bytecode works?

When we write a program in Java firstly, the compiler compiles that program and a byte code is generated for that piece of code when we wish to run this .class file on any other platform, we can do so. After the first compilation the bytecode generated in now run by the Java virtual machine and not the processor in consideration. This essentially means that we only need to have basic java installation on any platforms that we want to run our code on. Resources required to run the bytecode are made available by the Java virtual machine. which calls the processor to allocate the required resources. JVM's are stack-based so they stack implementation to read the codes

| Source Code (Program) | • Java file |

↓

(Compiler)

↓

| byteCode | • class file |

↓ ↓ ↓

(JVM (windows)) (JVM (Linux)) (JVM (Mac))

↓ ↓ ↓

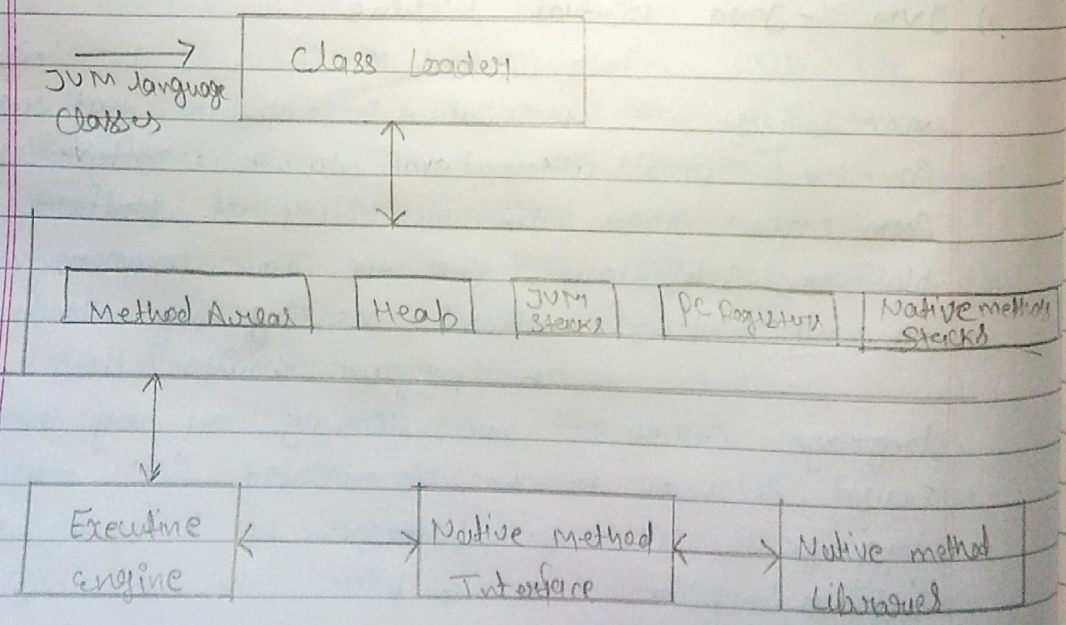| Machine Code | | Machine Code | | Machine Code |

2) JVM - Java virtual machine

      Java virtual Machine, or JVM as its name suggest is a "virtual" Computer that resides in the "real" Computer as a software process. JVM gives java the flexibility of platform independence platform independence that the same program works on any platform without needing any modification.

      A program written in a high-level language cannot be run directly on any Computer. First, it has to be translated into machine language. This translation can be done by a program called a Compiler. A Compiler takes a high level language program and translates it into an executable machine - language program.
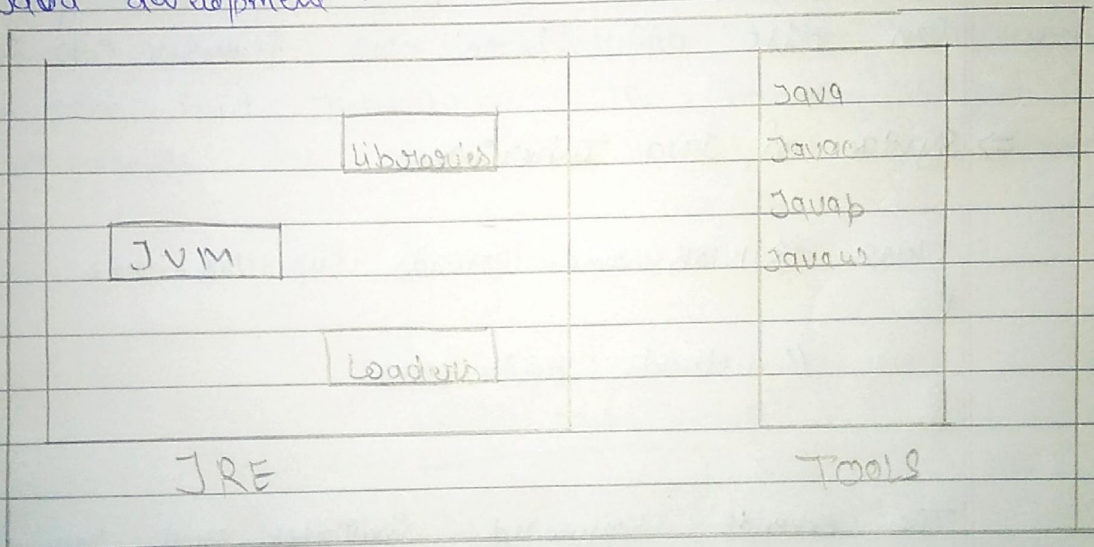
SHIVA KUMAR 20210004

Once the translation is done, the machine-language program can be run any number of times, but of course it can only be run one one type of computer.

There is an alternative to compiling a high-level language program. Instead of using a compiler, which translates the program all at once, you can use an interpreter which translate it instruction by instruction as necessary. An interpreter is a program that acts much like a CPU, with a kind of fetch and execute cycle. In order to execute a program the interpreter runs a in a loop in which it repeatedly reads one instruction from the program, decides what is necessary to carry out that instructions and then performs the appropriate machine language commands to-do so.

```
JVM language          ┌─────────────────────────┐
Classes  ───────────> │ Class Loader            │
                      └──────────┬──────────────┘
                                 ↕
┌──────────────┬────────┬──────────┬──────────────┬────────────────┐
│ Method Areas │ Heap   │ JVM      │ PC Registry  │ Native method  │
│              │        │ Stacks   │              │ Stacks         │
└──────┬───────┴────────┴──────────┴──────────────┴────────────────┘
       ↕
┌──────────────┐      ┌──────────────────┐      ┌──────────────────┐
│ Execution    │ <──  │ Native method    │ <──  │ Native method    │
│ engine       │  ──> │ Interface        │  ──> │ Libraries        │
└──────────────┘      └──────────────────┘      └──────────────────┘
```

3) JDK - Java Development Kit

Java Development Kit is a program development error environment that sits on top of the operating system layer as well as the tools and programming that developers need to compile, debug and run applications written in Java languages. It includes the Java Runtime environment (JRE), an interpreter / loader, a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.

JVM	Libraries		Java Javac Javap Javaws
	Loaders		
JRE			TOOLS

JDK - Java Development Kit