JVM: java virtual machine

It is used to change the byte code that is been generated by the complier into the language that is understandable by the processor.

* A program is written in java and complied by complier into byte code then converted into the language the is understand by the processor.
* But in other languages is the code is converted to object file that is understandable by the processor. This object file is considered as the input for the processor.
* JVM differs for the platform that is been downloaded.
* JVM is an interpreter that works between the complier and the processor so java code is more portable.
* In java interpreter that is JVM learns every byte code line by line and send it to the processor for execution. this makes the program slow in executing.
* Therefore, JIT [Just In Time Complier]is been introduced as a component inside JVM.



* It keeps all the repeated programs in ready as complied which becomes an easy task for the JVM for execution.
* JVM also consist of several libraries and files which are known as JRE [Java Runtime Environment].
* JDK is java development kit that contains javac that is java complier and other several tools but JDK consists of JRE itself entirely in them.
* IDE Integrated Development Environment that is for example Eclipse and Intellij IDEA.

METHODS:

What job is going to be done in an application is known as method. Like certainly some job is written inside the method and it is called whenever it is needed.

CLASSES:

When it is object oriented programming then definitely you will have class and object in them. It consists of the data and the methods related to the data are grouped inside them.

PACKAGES:

Classes are organised or grouped as packages in java. In java there are inbuilt packages like for square root and power of number.

* java.lang -java languages primitive datatypes and math functions
* java.io - input and output classes are organised
* java.util -all utility functions and date and time functions

**FILE NAME IS GIVEN AS THE MAIN CLASS NAME**