Introduction to Java (Java Terminology)

To begin Java learning, the learner must know the following terms:

Java Virtual Machine (JVM):

The JVM manages system memory and provides a portable execution environment for Java-based applications.

The JVM has two primary functions: to allow Java programs to run on any device or operating system (known as the "Write once, run anywhere" principle), and to manage and optimize program memory.

Technical definition: The JVM is the specification for a software program that executes code and provides the runtime environment for that code.

Everyday definition: The JVM is how we run our Java programs. We configure the JVM's settings and then rely on it to manage program resources during execution.

Bytecode in the Development process:

The Javac compiler of JDK compiles the java source code into bytecode so that it can be executed by JVM. It is saved as a .class file by the compiler.

Java Development Kit (JDK):

It is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc. For the program to execute in java, we need to install JDK on our computer in order to create, compile and run the java program.

JDK = JRE + development tools.

Java Runtime Environment (JRE):

JRE installation on our computers allows the java program to run. JRE includes a browser, JVM, applet supports, and plugins. For running the java program, a computer needs JRE.

JRE = Set of libraries + other files that JVM uses at runtime.

Garbage Collector:

The purpose of a garbage collector is to free the application developer from manual dynamic memory management. The developer is freed of the requirement to match allocations with de-allocations and closely take care of the lifetimes of allocated dynamic memory.

The garbage collector (GC) automatically manages the application's dynamic memory allocation requests.

A garbage collector performs automatic dynamic memory management through the following operations:

- Allocates from and gives back memory to the operating system
- Hands out that memory to the application as it requests it
- Determines which parts of that memory are still in use by the application
- Reclaims the unused memory for reuse by the application

ClassPath:

The classpath is the path that the Java Runtime Environment (JRE) searches for classes and other resource files.