Assignment

1) Components of JDK :

It stands for Java Development Kit & it is the software development environment which is used for developing java application.
JDK consists of components like - Loader, Compiler, an Archive file, a documentation generator with many other Components

2) Difference between JVM, JRE & JDK :

JVM - Stands for JVM (Java Virtual Machine) which provides Runtime environment for the Java programs.
In this Java appl^{n} can be executed

JRE - Stands for Java Runtime Machine used to execute Java program
JVM uses JRE for it's implementation

JDK - Stands for Java Development kit used for developing applications & it contains JRE & development tools

3) Role of JVM in Java ? How does the JVM execute Java Code :

JVM stands for Java Virtual Machine that compiles the Java program & produces the .class file.
JVM executes the code which includes load code, verify code, Execute code & provides the runtime environment.

4) Memory Management of JVM :

JVM has various data area which are used during execution of program.
Some Memory management done & Created by JVM & some are by thread.
Some of JVM memory Areas are Heap area, Method area, JVM Stack, Native Method Stack, PC Register.

5) JIT Compiler & it's role in JVM. What is bytecode & why it is important :

JIT Stands for Just In Time

It is runtime environment Component & used for the improrement of the Java Code or application. It Compile the Bytecode in the Machine Code at runtime.

At the runtime JVM loads the class files & determines the each individual of the bytecode & performs the necessary Computation

JIT Compiler is enabled by default. When the method is Compiled the JVM Calls the code directly & compilation does not take extra time & memory.

6) How JVM Works :

JVM Stands for Java Virtual Machine. JVM Calls the main method present in the Code. We can write java code once & can run it on any platform which is done by JVM & .class file is created.

Code is Compiled Step by Step & this Step by Step Compilation describes JVM

As Java follows the WORA Concept which means Write Once & Run Anywhere this is supported by the JVM.

Once the Code is written then it can be run on any device that has JVM.

The code is converted into the Machine understandable language through the compiler with the help of Java JVM so the code once written can be run on any operating System without any changes by JVM

Significance of class loader in Java & process of garbage Collection in Java:

Class Loader performs the task of Loading Java class into the JVM during runtime class loaders are also the part of JRE. JVM does not need to know about the files & files structure because of Class Loader.

Due to Garbage collector there is no need to allocate or deallocate the memory dynamically like C++ It's main purpose is to enhance Speed of application & memory leaks.