1) Explain the components of the JDK.

- The JAVA DEVELOPMENT KIT (JDK) is a software development environment used to develop Java applications.
Major components are
1. Java Compiler (Javac)
Java compiler is a key component of JDK that tranfers Java source code to byte code.
2. JAVA virtual Machine (JVM).
- Java virtual Machine is the running engine that executes Java byte code,
3. Java Runtime Environment (JRE).
- The JAVA Runtime Environment is a subset of JDK that includes the JVM and essential class libraries.

2. Difference Between JDK, JRE, and JVM.

→ JDK:- The JDK is a software development kit that develops applications in Java.

JRE :- The Java Runtime Environment is an implementation of JVM. It is a type of package that provides class libraries of Java, JVM, and other components.

JVM:- The JVM is a platform independent abstract machine that has three notions in the form of specifications.

3. What is the role of the JVM in JAVA? How does the JVM executes Java code?

- The role of JVM in java is that it enables a computer to run java programs as well as programs. Java code is first compiled to byte code and then generates a .class file, then the Java Virtual Machine interprets the class file for the underlying platforms. JVM is the one that actually calls the main method present in a Java code.

4. Explain memory management system of the JVM

→ The memory management system is responsible for dynamically allocation and managing memory resources for Java applications.

Heap memory:- ~~whatever objects created in~~ It is a big storage area where the java program keeps all its stuff.

Stack memory:- Stack Memory ~~is~~ is used for keeping track of what our program is currently doing.

Method area :- The Java program keeps information about its classes and methods in the method area.

Garbage collector :- Garbage collection is about getting rid of things you don't need anymore. ~~Garbage collect~~

Memory Tuning:- JVM provides options to tune memory parameters like heap size, garbage collector algorithms, and other related setting based on the applications requirement.

5) What are the JIT compiler and its role in the JVM? What is the bytecode and why is it important for Java?

- The JIT compiler helps improve the performance of Java programs by compiling bytecode into native machine code at run time.

- Bytecode is the set of intructions for Java virtual machine, It is executed when programs are written in the Java language.

6) Describe the architecture of the JVM.

→ The JVM is a software based emulation of physical computer that enables Java programs to run on any platform.

1. Class Loader:- Loads Java class file into the JVM's memory ~~and referenced by~~

2. Runtime Data Area.

Method Area: stores class metadata, static variables,

Heap:- Memory used for object and dynamically allocated memory.

Stack: Each thread has its own stack, storing method invocations.

PC Register:- Program counter registers keep track of the currently executed instruction for each thread.

Native Method stacks: for native methods a seperate stack used is maintained.

3. Execution Engine:- Interprets and executed bytecode instructions.

4. Native Method Interface (JNI) :- Allows Java code to call and be called by native applications and libraries written in other languages.

5. Native Method Libraries :- Contains native methods implemented in platform-specific code.

6. Garbage collector :- Manages memory by reclaiming memory occupied by objects that are no longer in use.

7) How does Java achieve platform independence through the JVM?

→ Java is platform independent because it is compiled to a bytecode that can be run on any device that has a JVM.

8) What is the significance of the class loader in Java? What is the process of garbage collection in Java.

→ Class loaders are responsible for loading Java classes dynamically to the JVM during runtime.

Garbage collection in Java is the automated process of deleting code that is no longer needed or used.