

- 1) The Java Development Kit (JDK) comprises three main components:
 - a) Java Compiler: It translates Java source code into bytecode.
 - b) Java Runtime Environment (JRE): It provides the libraries and runtime environment for executing Java applications.
 - c) Development Tools: These include tools like debuggers, profilers, and other utilities for Java development.
- 2) JDK (Java Development Kit) is a development kit for building Java applications. JRE (Java Runtime Environment) is an environment for running Java applications. JVM (Java Virtual Machine) is an abstract machine that provides the runtime environment in which Java bytecode can be executed.
- 3) The JVM (Java Virtual Machine) plays a crucial role in executing Java code. It interprets the bytecode generated by the Java compiler and executes it on the underlying hardware. It manages memory, handles exceptions, and provides various runtime services required for Java applications to run.
- 4) The JVM manages memory through automatic memory management, including heap and stack memory. It dynamically allocates memory for Java objects in the heap and deallocates memory for objects that are no longer referenced. The stack memory is used for method calls and local variables.
- 5) The JIT (Just-In-Time) compiler is part of the JVM that improves the performance of Java applications by compiling bytecode into native machine code at runtime. Bytecode is a platform-independent intermediate representation of Java code, which allows Java programs to run on any device or operating system that has a JVM installed.
- 6) The architecture of the JVM typically includes components such as class loader, runtime data areas (heap, stack, method area), execution engine (interpreter, JIT compiler), and native interface. These components work together to execute Java bytecode efficiently.
- 7) Java achieves platform independence through the JVM by compiling Java source code into bytecode, which is then interpreted or compiled into native machine code by the JVM at runtime. Since bytecode is platform-independent, Java applications can run on any device or operating system that has a compatible JVM.
- 8) The class loader in Java is responsible for loading classes into the JVM at runtime.

It locates and loads class files from the file system, network, or other sources. Garbage collection in Java is the process of automatically reclaiming memory occupied by objects that are no longer in use by the application. This process is managed by the JVM's garbage collector, which identifies and deallocates unreferenced objects to free up memory.