

Assignment - JKD

Date: / /

Que.1 Explain the components of JDK.

→ The JDK is a software development kit used for developing java application. It includes a variety of tools, libraries and components necessary for java development.

Here are the main component of JDK:

① Java Compiler (javac)

Primary tools which is used for compiling java source code (.Java file) into bytecode (.class file) that can be executed by JVM

② Java Virtual machine (JVM)

It is responsible for executing Java bytecode on diff. platforms.

③ Java Runtime Environment (JRE)

It provides the runtime environment needed to execute Java programs.

④ Java Standard library:

JDK includes a comprehensive set of libraries, APIs and classes that are provide core functionality for java applications.

⑤ development tools

The JDK includes various development tools that aid in development, debugging, and profiling of java application.

⑥ JavaFx

⑦ Additional tools and utilities.

Que. 2 Diff between JDK, JVM and JRE

→ JDK, JVM and JRE are three fundamental components of the java platform but they serve different purposes and have distinct roles in the java development and execution process.

- JDK is used for developing java application and includes development tools, libraries and JRE.
- JVM is abstract computing machine that executes java bytecode.
- JRE is subset of JDK and provides the runtime environment for executing java application without development tools.

Que. 3 what is role of JVM in java? How does the JVM execute Java code?

→ The JVM (Java Virtual machine) executes Java bytecode by loading, verifying, interpreting or dynamically compiling it into native machine code. It manages memory, handles exceptions and ensure platform independence by providing a consistent execution environment across different systems.

Que. 4 Explain the memory management system of the JVM.

→ The JVM's memory management system allocates and deallocates for

java object on the heap. It employs ~~or~~ garbage collection to reclaim memory from ~~was~~ unused object, using generational garbage collection to optimize performance. memory allocation is optimized through various techniques and developers can tune settings for better memory usage.

Que.5 ~~we~~ what are the JIT compiler and its role in the JVM? what is bytecode and why is it important for java? in short

→ ① JIT compiler (Just-In-Time compiler):

- The JIT compiler in JVM dynamically translates Java bytecode into native machine code at runtime.
- Its role is to improve the performance of Java application by compiling frequently executed bytecode into efficient native code.
- This optimization reduces interpretation overhead, enhancing execution speed.

② Bytecode

- Bytecode is an intermediate representation of Java code, generated by the Java compiler (Javac).
- It is platform-independent and can be executed on any system with a compatible JVM.
- Bytecode enables Java's "write once, run anywhere" principle, allowing Java programs to run on different platforms without modification.

Que.6 Describe the architecture of the JVM.

→ The JVM architecture consists of :

- ① class loader Subsystem : load classes into memory.
- ② Runtime Data Areas : memory Spaces like method Area, Heap, stack and PC Register.
- ③ Execution Engine :- Interprets bytecode and may use JIT Compilation.
- ④ Garbage collector :- manage memory allocation and reclamation.
- ⑤ Native method interface :- Allow interaction with native libraries.
- ⑥ Security Manager : Enforces Security policies.
- ⑦ Java Native Interface (JNI) : Facilitates integration with native code.
- ⑧ Java API : provides standard libraries and API's for Java development.

Que.7 How does Java achieve platform independence through the JVM?

- - Java achieves platform independence through bytecode and the JVM.
- Java programs are compiled into bytecode which is then executed by the JVM.
- The JVM interprets bytecode and translates it into native machine code at runtime, allowing Java applications to run on any system with a compatible JVM, regardless of the underlying operating system.

Que-8 what is the Significance of the class loader in Java? what is the process of garbage collection in java?

→

① Class loader:

- class loader are crucial in java for dynamically loading classes into the JVM during runtime.
- They allow java applications to load classes from different sources, such as the file system, network or memory.
- class loader provide flexibility and ~~flexibility~~ extensibility to Java applications by enabling dynamic loading of classes as needed

② Garbage collection:

- Garbage collection in Java is the process of automatically reclaiming memory occupied by objects that are no longer in use.
- The process involves identifying unreferenced objects, marking them as eligible for deletion and then deallocating their memory.
- Garbage Collection is transparent to developers, simplifying memory management and reducing the risk of memory-related errors.