

CORE JAVA REVISION:

Q1.What is the difference between Compiler and Interpreter?

A) **COMPILER:** It means java source code which High level language(HLL) as input and generates into Machine level language(MLL).

* It will scan High level language (HLL) code as only once.

INTERPRETER:It means java source code which high level language(HLL) and generates into Machine level language(MLL).

* It will scan High level language(HLL) as multiple times.

Q2.What is the difference between JDK, JRE, and JVM?

A) **JDK:** The term JDK means Java development kit, which means software having tools, libraries which developers to develop applications.

JRE: The term JRE means java runtime environment, Which helps execution java program having java virtual machine, class libraries and other components.

JVM:The term JVM means Java virtual machine, which understand the byte codes and also provide runtime for java programs.

Q3.How many types of memory areas are allocated by JVM?

A) There are Heap area, method area,JVM stacks,Native method stacks,Runtime constant pool,Direct Memory,

Q4.What is JIT compiler?

A)JIT stands for Just-In-Time. In the context of programming languages and virtual machines, a JIT compiler is a component that dynamically compiles code at runtime, as opposed to ahead-of-time (AOT) compilation where code is compiled before execution.

Q5.What are the various access specifiers in Java?

A)There are public, private, protected and default.

Q6.What is a compiler in Java?

A)Compiler compiles the code, when compilation successful it generates .class files, the .class files are used by JVM during execution.

In other words ,the source code HLL as input and generates MLL and The HLL code scans only once.

Q7.Explain the types of variables in Java?

A)In java, Mainly we discuss about two types of variables which are local variables and global variables.

Local Variables: The variables which are written inside a method is known as local variables.

Global variables:The Variables which are written outside method and inside class is known as Global Variables.

Q8.What are the Datatypes in Java?

A) **Datatypes in java:**

Byte(1 byte),short(2 bytes),integer(4 bytes),long(8 bytes),float(4 bytes),double(8

bytes),character(2 bytes),boolean(It size is machine dependent can't be predicted).

Q9.What are the identifiers in java?

A)*The identifiers in java which having name, which is for class name, method name,variable name,label name etc.,

- *The only characters allowed as identifiers in java.
- *The rules for identifiers a to z, A to Z,0 to 9 ,\$, _
- * Identifiers are starts with digits.
- * There is restriction on length of identifiers.
- * We can't use reserved words as identifiers.

Q10.Explain the architecture of JVM

A)The Java Virtual Machine (JVM) is a crucial component of the Java platform. It is responsible for executing Java bytecode,

which is the compiled form of Java source code. The architecture of the JVM consists of several key components that work together to provide a runtime environment for Java programs.

JVM Architecture:

1.Class Loader: The Class Loader subsystem is responsible for loading Java classes into the JVM. It performs tasks such as

locating and loading class files from the file system or network, verifying their integrity, and preparing them for execution. The class loader also maintains class identity, ensuring that duplicate classes are not loaded.

2) Class(Method) Area:Class(Method) Area stores per-class structures such as the runtime constant pool, field and method data, the code for methods.

3) Heap:It is the runtime data area in which objects are allocated.

4) Stack:Java Stack stores frames. It holds local variables and partial results, and plays a part in method invocation and return.

Each thread has a private JVM stack, created at the same time as thread.

A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes.

5) Program Counter Register:PC (program counter) register contains the address of the Java virtual machine instruction currently being executed.

6) Native Method Stack:It contains all the native methods used in the application.

7) Execution Engine:

It contains:

1 **A virtual processor**

2 **Interpreter:** Read bytecode stream then execute the instructions.

3 **Just-In-Time(JIT) compiler:** It is used to improve the performance. JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here, the term "compiler" refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

8) Java Native Interface:Java Native Interface (JNI) is a framework which provides an interface to communicate with another application written in another language like C, C++, Assembly etc. Java uses JNI framework to send output to the Console or interact with OS libraries.