

Q1.What is the difference between Compiler and Interpreter?

Ans: A Compiler takes a program as a whole. An Interpreter takes single lines of a code. The Compilers generate intermediate machine codes from High level code. The Interpreters never generate any intermediate machine codes.

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

Ans: JDK stands for Java Development kit, it provides libraries and the required files to run our java programs.

JDK => JRE + JVM

JRE -> Java Runtime Environment, it provides suitable environment to run our java program.

JVM :: Java Virtual Machine, It is responsible to run our java programs on the basis of Multithreading.

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

Ans: The memory in the JVM is divided into 5 different parts: -

1. Class (Method) Area

The class method area is the memory block that stores the class code, variable code(static variable, runtime constant), method code, and the constructor of a Java program.

2. Heap

The Heap area is the memory block where objects are created or objects are stored.

3. Stack

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

4. Program Counter Register:

Each JVM thread that carries out the task of a specific method has a program counter register associated with it.

5. Native method Stacks:

Also called C stacks, native method stacks are not written in Java language. This memory is allocated for each thread when it's created.

Q4:What is the JIT compiler?

Ans:The Just-In-Time (JIT) compiler is a component of the runtime environment that improves the performance of Java™ applications by compiling bytecodes to native machine code at run time.

Q5:What are the various access specifiers in Java?

Ans:Java provides four types of access modifiers or visibility specifiers i.e. default, public, private, and protected.

Q6.What is a compiler in Java?

Ans: It is a software which takes sourcecode(HLL) as the input and generates MLL code as the ouput.To convert the HLL code to MLL code compiler will scan the HLL code only once.

Q7.Explain the types of variables in Java?

- Local variable: These variables are declared and used within a specific method, constructor, or block of code. They are only accessible within the scope in which they are defined. Local variables must be initialized before they can be used.
- Instance variable: These variables are declared within a class but outside any method, constructor, or block of code. They are associated with individual instances (objects) of the class. Each instance of the class has its own copy of the instance variables. Instance variables are initialized to their default values if not explicitly assigned.

- Static variable: These variables are declared with the **static** keyword within a class but outside any method, constructor, or block of code. They are associated with the class itself rather than with instances of the class. Static variables are shared among all instances of the class, and changes to the variable will be reflected in all instances. Static variables are initialized to their default values if not explicitly assigned.

Q8.What are the Datatypes in Java?

Ans: It specify that which type of value we stored in a variable.

- Primitive data types - includes byte , short , int , long , float , double , boolean and char.
- Non-primitive data types - such as String , Arrays and Classes.

Q9.What are the identifiers in java?

Ans: Identifier is a name in java program. It can be class name ,method name ,variable name ,label name.

=> The only character allowed for java identifiers is

=> a to z, A to Z,0 to 9,\$,_

Can identifier start with digits? No

Is there a restriction on the length of identifiers? No

We can't use reserve words as an identifier. We can use inbuilt class names and variable names as identifiers, but it is not a good

practice to use.

Q10.Explain the architecture of JVM?

Ans: The JVM consists of three distinct components:

1. Class Loader
2. Runtime Memory/Data Area
3. Execution Engine

Class Loader

When you compile a .java source file, it is converted into byte code as a .class file. When you try to use this class in your program, the class loader loads it into the main memory. The first class to be loaded into memory is usually the class that contains the main() method. There are three phases in the class loading process: loading, linking, and initialization.

RunTime Area: It consist Stack area ,Method area, Class area,ProgramCounter,Naivestack area.

Execution engine: It consist JIT compiler, Interpreter, Garbage Collector.