JDK Assignment

1] Explain the components of JDK.

Ans) The components of JDK:

I] Java Compiler (Javac):

i] The compiler converts Java Source code into bytecode, which is platform-independent & can be executed by the JVM.

II] Java Virtual Machine (JVM):

i] A key component of java runtime envirorment that executes java ~~code~~ bytecode.

ii] It provides a platform-independent executio enviorment that converts bytecode into machine-specific instructions.

III] Java Runtime Enviornment (JRE) →

i] Includes the JVM, core classes, & supporting files.

ii] It provides libraries, JAVA API's, & components needed to run java applications & applets.

IV] JAVA API (Application (programming Interface)

i] A large collection of ready-made software components that provided useful ~~&~~ capabilities. such as GUI components, networking, I/O.

Page No. | Date

v] security features → Include tools & API's for securing java application such as key & certificate mangement utilities.

2] Diffrentiate bet^n JDK, JVM & JRE

ns. I] JDK :—

i] JDK is software development enviornment which is used to develop java development.

ii] Contains JVM & interpreters & loader, javac, jar file to complete the development of Java.

II] JVM →

i] JVM is an abstract machine.

ii] it doesn't have physical existence.

iii] provides runtime enviornment to execute java bytecode

IV] JRE →

i] It is for java Runtime enviornment

ii] It is implementation of JVM.

iii] It Physical exists.

iv] Contain set of liberaries & other file

3] What is the role of JVM in java? How does the JVM execute Java Code?

Ans. I] Role of JVM in Java →

i] The JVM is responsible for executing java bytecode

ii] JVM allows java app to run on any device

iii] provides a secure execusion envionment.

II] How does JVM execute JAVA code.

i] The java source code is compiled into bytecode.

ii] verifies the bytecode & load the java program th^2 class loader into JVM memory.

4] Explain the memory management system of the JAVA.

Ans. → i] JVM manages memory automatically using a garbage collector (GC).

ii] Heap → The largest memory area in JVM, where object & array created during program execution are stored.

iii] Garbage Collector → it pauses all other process.

iv] stack → store method calls, local variable.

5] What are JIT compiler & its role in JVM? What is the bytecode & why is it important for java?

Ans. i] JIT compiler is component of java Virtual Machine that dynamically compile bytecode into native machine code at runtime.

ii] It improves the performance of java application.

iii] compiled native code generated by the JIT is cached in memory to avoid recompilation.

iv] Bytecode is low-level representation of java code executed by JVM.

v] When you compiler java source code file, the java compiler translates into bytecode

6] Describe the architecture of JVM.

Ans. 1] Classloader →

i] Bootstrap Classloader → this is the classloader which is super class of extension classloader. It loads rt.jar file which contains all class file.

2] Extension ClassLoader → it is child classloader of Bootstrap.

2] Class Area →

i] class area stores per class structure such as runtime constant pool.

3] Heap → runtime data area in which objects are allocated.

4] Stack → Java stack stores frames.

5] program Counter Register → it contains the address.

7] How does Java achieve platform independen through the JVM?

Ans. 1] The meaning of java platform-independent is that the java compiled code (byte code can run all operating system.

2] the main point here is that role the JVM depends on the operating System - so if you are running Mac OS X you will have diffrent JVM than if you are running Windows or some other operating System.

→

8] What is the significance of the class loader in Java? What is the process of garbage collection in java?

Ans. 1] The class loader is part of JVM that loads classes into memory.

2] It is responsible for finding & loading class file at runtime.

3] The class is critical part of the Java runtime enviornment.