STUDENT ID - 24034052000S

PAGE NO.:
DATE: / /

Assignment - 02.

Que.1. Explain the components of JDK.

Soln. JDK contains the following components.

- Javac - It represents the Java compiler which converts source code into Java bytecode which helps in system to understand the user written code.

- Jar - Java archive, It includes libraries packages, class into a single Jar file.

- Java - Java is a loader for Java applications. It interprets the class files generated by compiler.

- Jstack - It is used to trace Java threads ~~using jcmd also..~~

- Javap - It disassembles the class file.

Que.2 Differentiate between JDK, JVM and JRE.

JDK - It is a Java Development kit which contains libraries, class, packages. It is a storage of all required files and libraries in Java.

JVM - Java virtual Machine host class files means it reads class files and execute .class files and runs program.

JRE - Java runtime Environment is used by JDK tools and it includes essential components needed to run Java applications.

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

Soln Role of JVM in Java is to read class files in compiler and perform the operations. ~~like checking/ checking datatypes~~.

Execution of Java code.

- Code is compiled into bytecode.
- Load the program through class loader into JVM memory.
- Initialize JVM instructions & data.
- Initialize the value inside program counter.
- Execution process.

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

Soln During the Runtime of Java code memory managment system allocates ~~and~~ or deallocates memory space for variables & objects.

To make memory available for new objects.

Memory management is the process to remove unused objects from heap. (it is an area where objects reside).

Garbage collection frees space in the heap for allocating new objects. After collection compaction takes place to sort out free space created in memory.

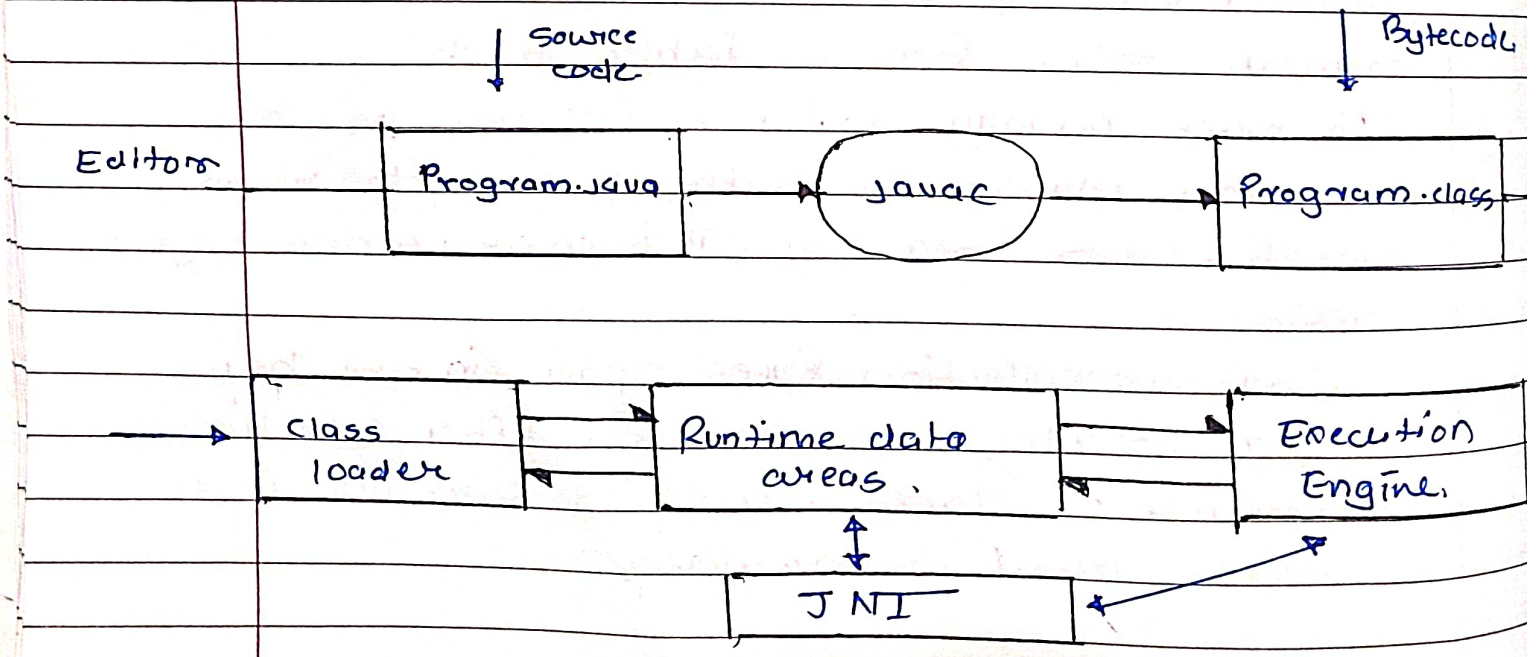
Que 5 What are the JIT compiler and its role in the JVM? What is the bytecode and why is it important for Java?

Soln - JIT compiler is a component of runtime environment that compiles bytecode to native machine code at runtime.
The role of JIT is to improve performance of Java program.
Just in Time compiller is important because it needs less memory usage and JVM directly calls compiled code directly instead of interpreting (interpreter code line by line).

Que.6. Describe architecture of JVM.

Soln JVM architecture helps in Running the Source code of Program after compilation and reading class file.

Major components of JVM architecture contains -



Components -

1. Class loader.
   - Bootstrap loader
   - System loader.

   Its Job is of loading classes and generate data that constitutes defination for a class.

2. Runtime data Areas.
   - Heap
   - Method area
   - Java stack.

   They are used during execution of program. Some of these data areas can only be destroyed when JVM exits.

3. Execution Machine Engine
   - JIT
   - Garbage compiler.
   - It is responsible for execution of bytecodes loaded in JVM.

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

Soln. Java achieves platform independence when it compiles to a bytcode which can run on any device which has a JVM. This means that you can write a Java program at one platform and then run it on different platform this is because JVM converts source code to bytecode which is 'independent platform. class file.

DATE: / /

Que-8 What is the significance of class loader in Java? What is process of garbage collection in Java?

Soln A class loader is an object responsible for loading classes. Its job is to transform the name into file name and read a 'class file' of that name from a file system.

The Garbage collection is a process to free up space in memory to allocate new objects.

Steps.

1. Object Allocation - Any new objects are allocated to eden space. Both survivor spaces starts getting empty.
2. filling the eden space - A minor garbage collection takes place when eden space is filled
3. Copying Reference objects, Reference objects are moved to first space and rest is deleted.
4. Aging.
5. Compaction - To create more space compaction takes place and objects are being sorted out.