ASSIGNMENT NO : 3

Q1 Explain the components of the JDK.

→ • Java Compiler : Translates java source code into bytecode for execution by the Java Virtual Machine (javac)

• Java Runtime Environment (JRE) : Provides the necessary libraries and JVM to run java applications including tools like java, javap

Q2 Differentiate between JDK, JVM, JRE

→

	JVM	JRE	JDK
Purpose	executes java bytecode and provides platform independence	provides runtime enviroment to execute java applications	provides tools for java development and compilation
Includes	JVM runtime, garbage collector, class loader	JVM, core lib, supporting files	JRE, development tools.
Scope	core component used by JRE & JDK	used for running java applications	used for developing, compiling & running java applicat^{n}

Q.3 What is role of JVM? How does JVM execute java code?

→

- Java Virtual Machine executes java bytecode, providing a platform-independent environment for running java applications. It manages memory, handles garbage collection and ensures that java programs can run consistently across different Os.
- 1> Loads java bytecode from .class file into memory.
  2> Then it verifies bytecode.
  3> Executⁿ :- a) It interprets bytecode instruction executing them one by one.
  b) It uses JIT compilation (Just In Time) to convert frequently executed bytecode into native machine code at runtime.
  4> Memory Mngment :- allocatⁿ & deallocatⁿ, garbage collection, etc.

Q.4 Explain Memory Management system of JVM.

→ Java memory mngment systᵐ allocates memory for objects & uses garbage collectⁿ to automatically reclaim memory from objects that are no longer in use, minimizing manual memory mngment. It involves heap managment for dynamic memory allocatⁿ & stack mngment for method calls & local variables.

Q.5 What are JIT compiler & it's role in the JVM? What is bytecode & why is it important for Java?

→ • It improves performance by compiling frequently executed bytecode into native machine code at runtime. This reduces overhead of interpreting bytecode, leading to faster executⁿ of Java applicatⁿ.

• Bytecode is intermediate, platform-indepdent code generated by java compiler that JVM executes. It is important as it provides platform independent java code.

Q.6 Describe architecture of JVM.

→ • It has :- a) Class loader Subsystem - loads & verifies java classes

b) Runtime Data Area - includes heap, stack & method areas.

c) Executⁿ Engine - interprets bytecode / uses JIT to execute java prog.

d) Native Interface - allows java code to interact with native applicatⁿ & lib.

Q.7 How does java achieve platform indep. through JVM?

→ It achieves by producing bytecode which can be executed anywhere through JVM.

Q.8 What is the significance of the class loader in Java? What is the process of garbage collection in Java?

→ • It dynamically loads Java classes into JVM at runtime, ensures security and consistency of code.

• Garbage collectⁿ:

i) marking : identifies unreachable objects.
ii) Normal Deetn : unreachable obj made eligible for deletn
iii) Delete with compactn : heap compacted before deletn.
iv) finalizatn : run finalizers / cleanup methods.

Q.9 What are four access modifiers in Java, and how do they differ from each other?

→ (i) public : accessible from anywhere
(ii) protected : within same package & subclasses
(iii) default : same package (package private)
(iv) private : accessible within same class.

Q.10 What is difference betn public, protected & default access modifiers?

→

Q.11 Can you override a method with a different access modifier in a subclass? Explain.

→ NO. To maintain consistency and usuability, it is not allowed. It leads to unexpected behaviour.

Q.12 What is difference between protected and default (package-private) access?

→

(i) protected : accessible within same package & also in subclass, even if subclass is in different package.

(ii) default : accessible only within same package.

Q.13 Is it possible to make class private in Java? If yes, where can it be done, & what are the limitations?
→ Yes. Private class are not accessible outside class.

Q.14 Can a top-level class in Java be declared as protected or private? Why or Why not?
→ No. Does not $\overset{\boxed{\text{allow}}}{\vee}$ inheritance.

Q.15 What happens if you declare a variable or method as private in a class & try to access it from another class within the same package?
→ Compilation Error. The private variable or method are not accessible outside class.

Q.16 Explain concept of 'package-private' or 'default' access. How does it affect visibility of class members?
→ • It makes class members accessible only within the same package. It restricts visibility to other classes within same package but hides members from classes in different packages.

—•—x—•—x—•—x—•—x—•—x—