Assignment 3 (419124)

Q1

→ The JDK is software development kit used to develop Java applications It provides the JRE Java Runtime Environment which has libraries, JVM Java virtual machine and other components to run a Java application with development tools like Java compiler, docs generator, debugger

Q2

→ JDK is a development kit which includes development tools and libraries JVM is Java virtual machine which executes Java bytecode and provides platform independence. JRE which stands for Java Runtime Environment contains JVM and necessary libs to run a Java app

Q3

→ JVM is responsible to execute bytecode and to provide runtime environment JVM translates bytecode into machine code through the interpreter or through JIT compiler executing it to a machine

Q4

→ In JVM is different memory management areas Heap (for objects), Stack (for methods execution), method area (for class level data) and Garbage collection which free up memory and prevents memory leakage

Q5

→ The JIT compiler is a improves performance by

Interpreting executed bytecode into native machine code at runtime. Bytecode is the intermediate code generated by the Java compiler

Q6.

→ The JVM consists of class loader subsystem, Runtime data Area, and Execution Engine

Q7.

→ Java programs can be ~~run~~ executed can any machine so it is platform independent. Java source file is converted into the bytecode which is not machine-specific. The JVM which is available for different platforms, ~~interpreter~~ executes the bytecode allowing Java program to run on any machine.

Q8.

→ The class Loader in JVM loads Java classes at runtime which enables dynamic loading. Garbage Collection automatically manages memory by removing objects that are no use which optimizes the memory.

Q9.

→ There are four Access modifiers public, private, protected, ~~class~~ package level private. Public allows access from any class, private restricts access to the defining class, protected allows access within the same ~~class~~ package or subclasses and package level private permits access only within the same package.

Q10.

→ Public members are accessible from any class, protected members are accessible within same package and subclasses. default members are only accesible within the same package.

Q11.

→ We can not override a method with a more restrictive access modifier. A protected method in a superclass cannot be overridden with a private method in a subclass, as it violates the visibility requirements.

Q12.

→ Protected allows access within the same package and subclasses even in different packages, default access restricts to the classes within the same package.

Q13

→ A top-level class cannot be declared as private, only inner classes can be private, restricting their accessibility to the outer class only.

Q14

→ A toplevel class in Java cannot be declared as protected or private. Only public and package-level private are allowed because protected and private are meant for restricting member access, not classes.

Q15

→ Private members of a class cannot be accessed from another class, even within the same package.

An attempt to do so will lead to compilation error.

Q16

→ Default access can only be accessed by classes within the same package. This access level is neither too restrictive like private nor too permissive like public, promoting package level encapsulation.