Module 2:- OOPJ.

Assignment 3.

1) Explain components of JDK.
   i) Java Development tools
   ii) Java API docs
   JRE { iii) Java supporting libraries. (rt.jar)
   JRE { iv) Java execution environment (JVM).
   v) Code samples (src.zip)

2) Differentiate betn JDK, JVM and JRE.
   JDK:- It includes JRE & JVM. It provides the environment to develop & execute java program.

   JRE:- It allows only running of program. It doesn't develop.

   JVM:- It is the actually the one, who is responsible for executing the program line by line It acts as interpreter.

3) Which are four access modifier?
   i) public
   ii) private
   iii) protected
   iv) default.

4) Difference betn public, protected & default access?

	public	default	protected	private
class	✓	✓	✓	✓
package	✓	✓	✓	
outside package Subclass	✓		✓	
outside package.	✓			

Rainbow

5) Diff. between protected & default.
protected modifier can access the data outside the package using inheritance. but default or package-private modifier can't access data outside package.

6) What happens if you try to access private variable or method from another class with same package.
— We cannot access it, because private means only for that particular class, we cannot access out of it.

7) Explain concept of package-private or default access.
— Default modifier has the access only within the package. It cannot be called outside the package.

8) Is it possible to make a class private in Java. where it can be done.
— Yes we can make a class private but only only inner or nested class.

9) Can a top-level class in Java be declared as protected or private? why or why not.
— Toplevel class cannot be declared as ~~prot~~ private, because it would be completely useless as nothing would have access to it.

bow

③ OECD - 1961

Paris, 38 सदस्य

③ ECTA- Economic Cooperⁿ & Trade Agreement

Ind-Aus (FTA) (29 Dec. 2022)

10) What is role of JVM in Java? How does Java execute Java code?

— JVM runs java applications. JVM is the one that calls main method present in Java. It is part of JRE.

When we compile the code of java file class files having byte code are generated by Java compiler. This class file goes into various steps when we run it. nothing but JVM.

1) Class loader system :- This reads class files generate binary data & save it in method area. After loading, JVM creates an object of class to represent this file in heap memory.

2) Memory Areas :-

3) Execution engine :- This executes byte code line by line, uses data and information present in various memory area.

Interpreter

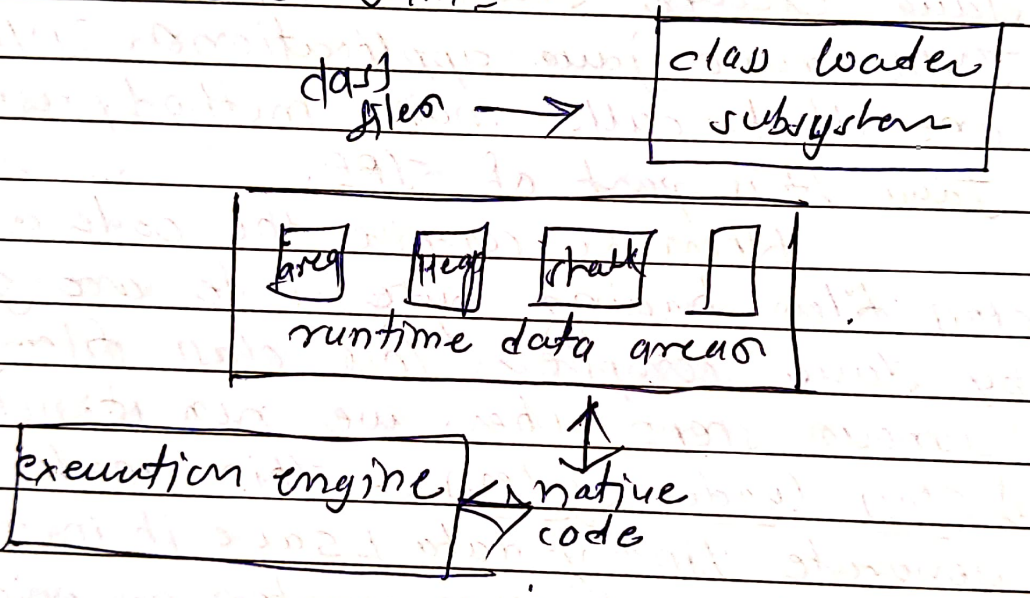
JIT (just in time compiler) - It compile byte, and changes it to native code so whenever interpreter sees repeated method calls, JIT provides direct native code.

11) What are JIT compiler & its role in JVM?

→ Just in Time compiler is used to increase the efficiency of interpreter. When interpreter sees any repeated method calls, JIT gives directly native codes, so it avoids calling interpreter every time.

JIT compiler converts byte code into native code.

12) architecture of JVM:



13) How does Java achieve platform independent through JVM.

→ Write once, run anywhere give JVM a platform independancy. Java source code is compiled into a bytecode, which is platform-neutral. This bytecode can be executed on any platform that has JVM compatibility with bytecode. regardless of operating system or hardware.

14) what is the significance of the class loader in Java? what is process of garbage collection

— Class loader is key component of JVM, that dynamically ~~loads~~ loads Java classes into JVM at runtime. JVM does not need to know about files & file system because of loaders.

Garbage collection is automatic process in Java that allocates & deallocates memory without need for explicit programming. To avoid out of memory errors, garbage collection is the solution. This deallocates memory of unused instances/objects present in Heap memory.

15) Explain memory management in Java.

— main aspects are :- i) JVM memory structure
ii) Garbage collector

JVM define runtime data area which are used during an execution.

JVM contains :- Heap area, Java stacks, method area, PC registers & Native method stack. Java itself do the memory management.

16) Can you override a method with different access modifier in a subclass? ex. protected method of superclass can be overriden with private or default in subclass.

— Yes, protected method of superclass can be overriden by subclass. but only if subclass method have protected or public access modifier which means subclass overriden method should not have a weaker access modifier.

ubow