Assignment

Explain

1] components of JDK.

→ The main components of JDK (Java develope -ment kit) are

Java ~~SDK~~ includes, Java Developement tools, Java API Tools, rt.jar, files, JVM machiner.

* rt.jar files are API libraries.

2] Differentiate bet^n JDK, JVM, JRE.

i) JDK - It is developement kit for developer or programmer to develope project in java. but we need JVM for this.

ii) JVM it is enviornment/platform on which we can use JDK kit to develop project

iii) JRE → Java Run time enviornment which ~~need~~ needed to run java application Java runtime envirmet include JVM machin as well as rt.jar files which are library filer.

rt.jar contains .class filer.

JRE provided only envidrment to run java but not tools/kit for java applicatio

iv) Unlike JRE which provided enviornment and JDK which provide tools for java programm
JVM → used to execute compiles files on the given platform - It also performs memory management task in the platforms. It carry out task form loading class files to execution.

Q.3] Role of JVM in Java.
JVM is used to execute to java code by using Compiler loader and execution engine.
It is required to have Java virtual machine to successfully run java.
Components of JVM →
i) class loader subsytem : It loades class files in O-S which are libraries.
ii) execution engine : execute bytecode info.
It includes interperter and run-time compiler which converts byte code to native machine code.
iii) Runtime Data areas:-
method area, heap, stack, PC registers, native method stacks
Heap: objects and their data
Stack: local variables, method calls
method area: store class informat, class fields, parameter type, access modifiers

Q.4] Explain memory management in JVM:
→ memory management in JVM is done by runtime data areas.
i) method area: Store class information, class fields, access modifiers, class field

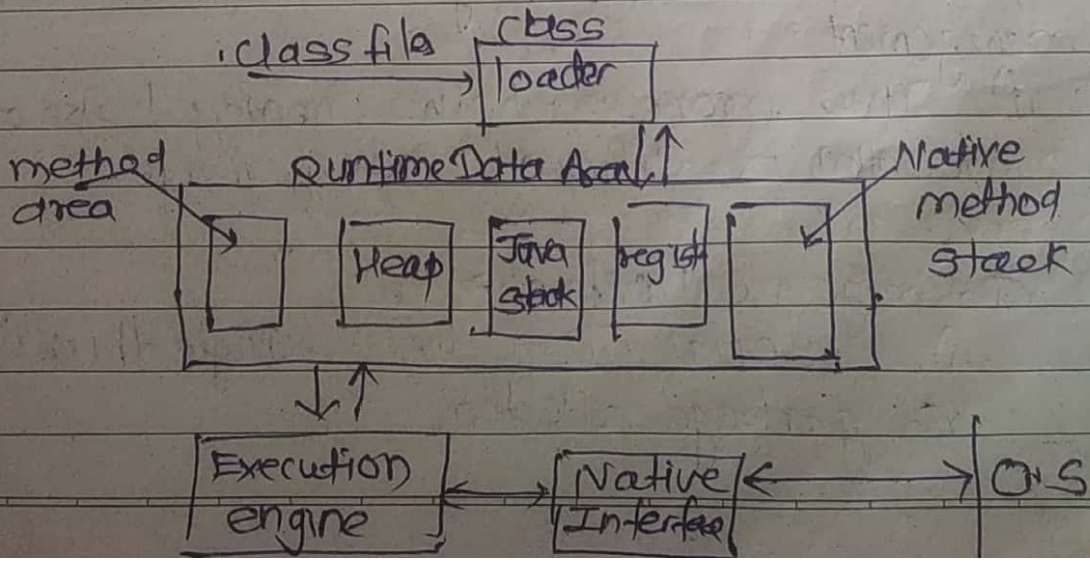
ii) Heap - stores objects and their bytecode.
iii) Stack - stores & process local variables, method calls.
iv) PC-registers → addresses of bytecode to be execude to track control flow of java code.
v) Java Native interface ⇒ allows java code running in JVM to be incorporated in C/C++.

Q.5] JIT compiler ⇒ & its Role

i) It is compiler on JVM for compilation of Java .class files.
ii) It converts bytecode to native code.
iii) This execution occurs at runtime hence also called as runtime compiler.

Initial Execution → JVM interprets bytecode
JIT compilation → JIT compiler compiles it to native code.
Optimized Execution → subsequent calls to use compiled native code, resulting in faster execution.

Q.6] Architecture of JVM :-



Q.7] Java platform independence through JVM?

→ i) Java achieve platform independence through JVM.

a) JVM compiles Java files .class files during execution. (It is in bytecode and independent of system.

b) JVM is a layer betn Hardware (O.S) and Java program bytecode (.class files), it translate this code into native machine code which is platform dependent

c) Java acheives platform independence because same .class files can be run on other system or platform.

As Java .class files are portable they give independency to JVM.

Q.8] Significance of class loader →

i) class loader is component which loads java files into memory.

ii) It allows .class files to be loaded for execution on demand

iii) Class loder is responsible for loading linking, intialization of java class files

Garbage collection in Java →

i) It is done for memory allocation and memory management.

ii) This process helps memory leaks prevention & efficient memory usage

Garbage collectors →

i) Serial GC → uses single thread for garbage collection.

Parallel GC → uses multiple threads to perform garbage collection

Concurrent garbage collector – performs garbage collection continuously with operation

G1 (garbage first garbage collector) → Aims to balance throughput & pause time by dividing heap into regions & prioritizing the collection of garbage

Q.9] Access modifies in Java

Access modifiers determine visibility of class methods & variables

1) Public – class is accessible from any other class.

eg Public static void main (String [ ] args) {

2) Protected – class member is accessible within its package or by its members

eg Protected int value;

3) default – class is only accessible within its own package

eg class Value {
int value;
}

|| default → package level private

4) private :– visibility of class members is accessible with its same class and not outside the class.

Assignment 3.

Q 10] difference between public, protected, default acess modifiers.

→ i) These are all Acess modifiers which shows visibility of members of class/enum/ interface.

ii) Public modifier → Acessible within its same class as well as sub classes and outside its package too.

iii) Protected → Acceessible with its package class, sub class, other classes as well as non package sub class

iv) default Acess modifiers → Acessible wit its same package only ie. same classes, sub class & other classes in same pack

Q.11] Can you override a method with different access modifier within subclass?

⇒ In order to override a method with a diffent access modifier within subclass,

The access modifier of overriding method must have same access modifier or it should not have more protected access modifier as compare to its superclass.

i) A protected method in superclass can be overridden with access modifier subclass of protected or public.

ii) we can not use private access modifier subclass to overide the superclass because private access modifier are accessible within its own class only.

Q.12] Difference between protected and default modifier ?

⟹

i) Protected access modifier:

protected access modifier ~~is accessible~~ is accessible within its same package class, sub class & other class, Also it include members of other sub class of non package class

ii) Default access modifier ⇒ default access modifier controls visibility of members of same class package members which include class, its sub-classes, non sub class.

Q.15] What happens if you declare a variable or method as a private in a class and try to access it from another class within same package?

→

In private access modifier → variables declared are accessible within its class only and other classes within same package can not access it.

So if we need provide access to variable we can use protected access or package level access modifier which will give access to other classes also.

Q.16] explain concept of package level private or default access. How does it affect visibility of class members.

⟹ i) For default access modifier the members of class / enum / instances are accessible to other classes in same package class and not to other classes outside its package.

ii) Scope of ~~pair~~ default access modifiers is limited within its same package.

iii) By using default access modifiers we can control accessibility of your class members and implementations in the class will be unaccessible from other classes. which will give better and modular code in java.