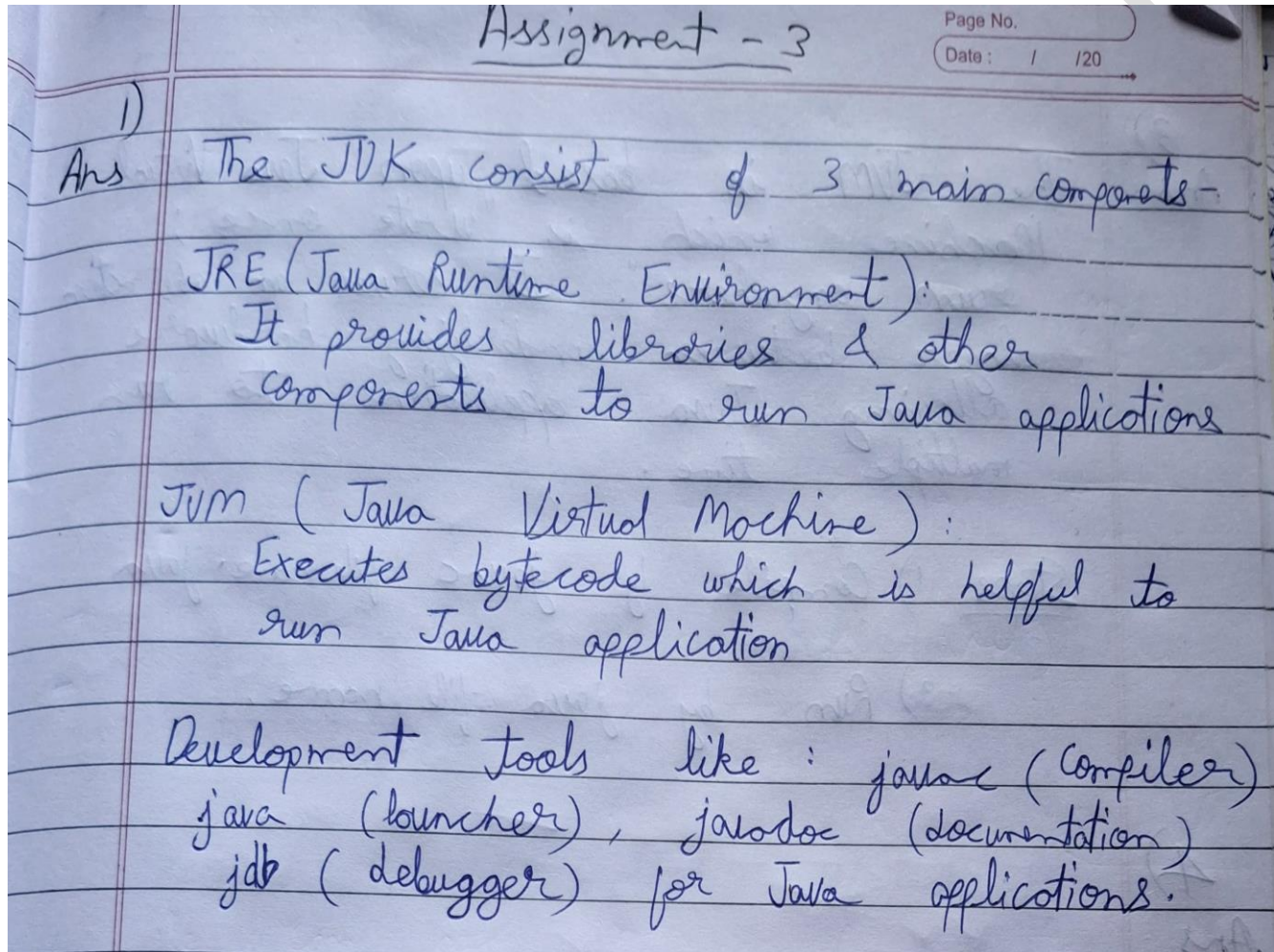


CDAC Mumbai PG-DAC AUGUST 24

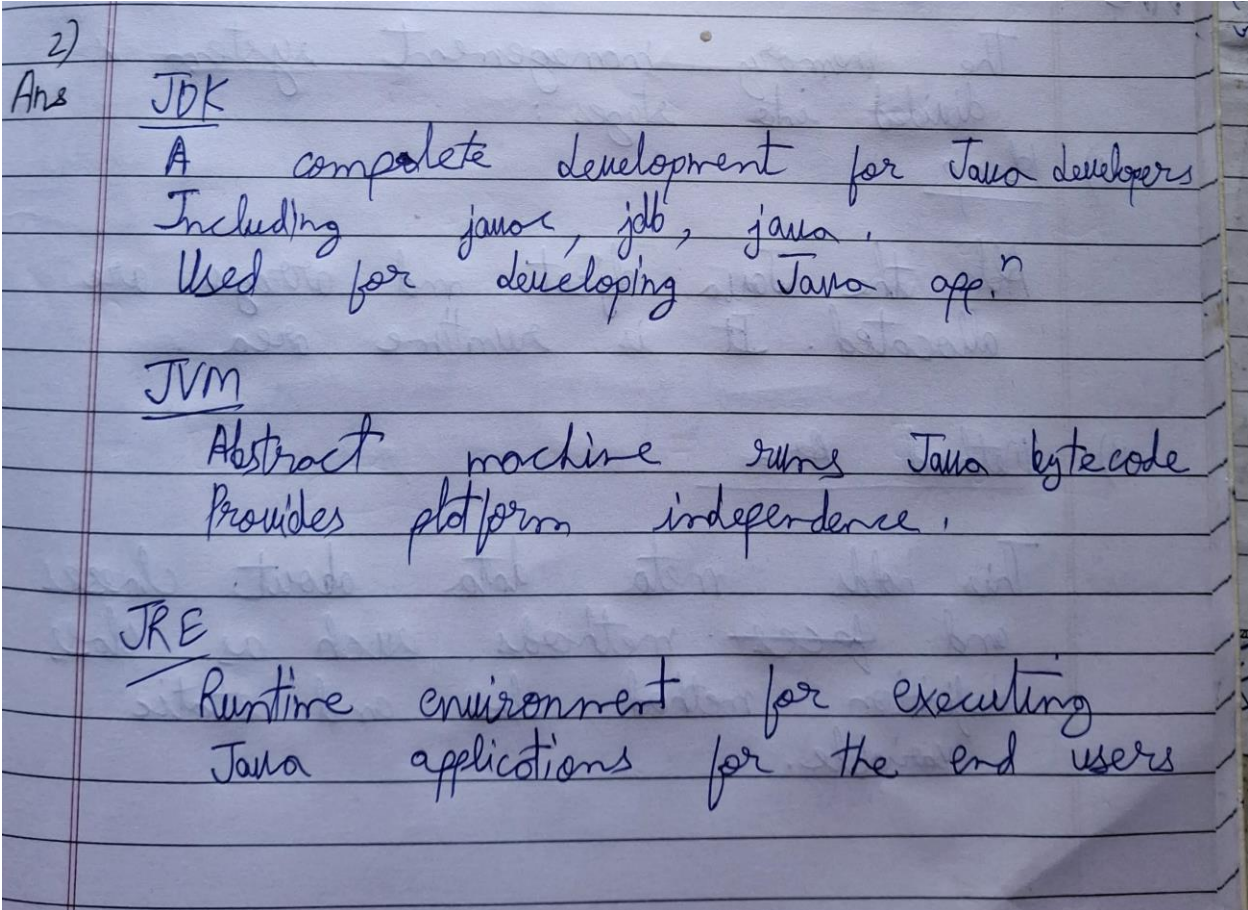
Assignment No- 3

Note: Write down this Interview questions & answers in your notebook.
Take screenshots, make word file & upload on Github.

- 1) Explain the components of the JDK.



2) Differentiate between JDK, JVM, and JRE.



3) What is the role of the JVM in Java? & How does the JVM execute Java code?

3)
Ans

The JVM is based upon Java Virtual Machine which is "write once; run anywhere". It serves an abstraction layer between underlying hardware allowing Java applications to run multiple times.

Steps 1) Compile by `javac -filename.java`

2) Run by `java -file name,`

4) Explain the memory management system of the JVM.

4)
Ans

The memory management system is divided into stages:

i) Heap

All the Java object and arrays are allocated. It is runtime area.

ii) Method Area

This holds meta data about classes and ~~faces~~ methods such as class definition, method, class and static variable.

iii) Java Stacks

Each thread has its own Java stack store from representing method calls and local variable.

iv) Program Counter (PC) Register

It holds the address of current instruction being executed by thread.

v) Native method Stack

Executes native method with stack.

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

5)

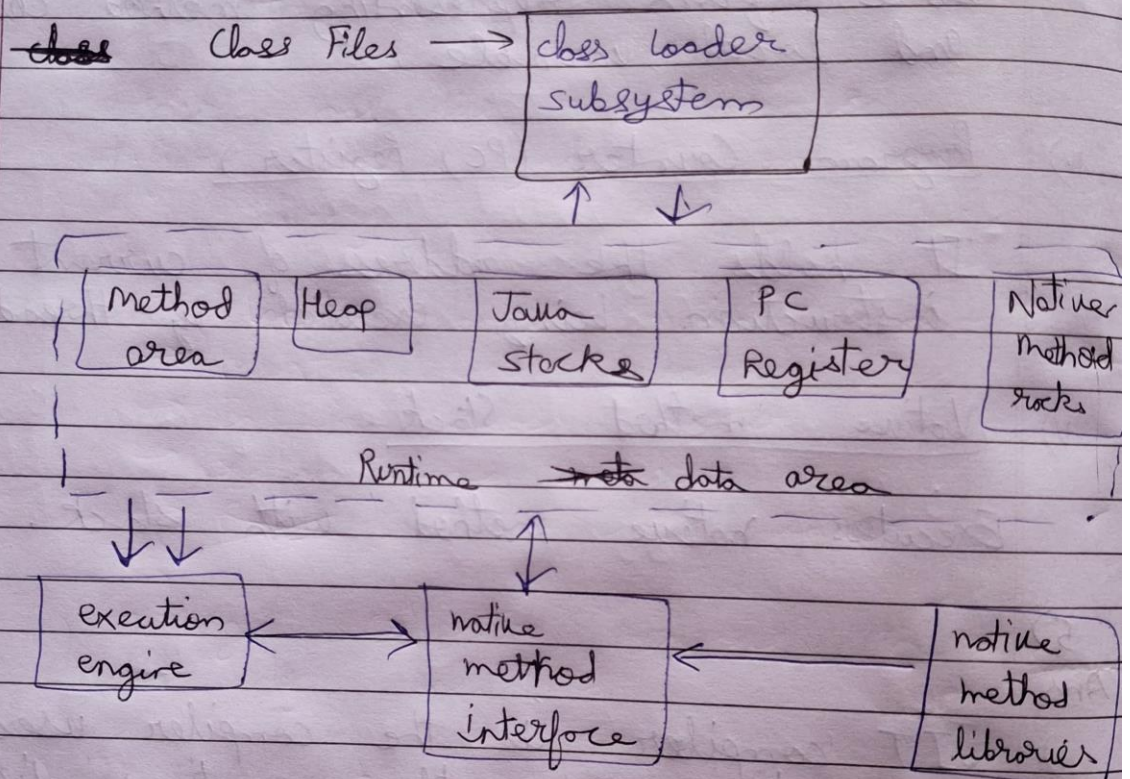
Ans

- i) JIT compiler is the compiler used in JVM for frequently executings like execution.
- ii) It is used for performance enhancement of execution engine.
- iii) Byte code is code after compilation of java file & .class file.
- iv) The JIT compiler convert bytecode to the native code.
- v) It is not OS dependent and can run on any OS and binary file.

6) Describe the architecture of the JVM.

6)
Ans

Architecture of the JVM



There are 3 main components of JVM-

- i) Class loader subsystem
- ii) ~~Outline~~ Runtime data area
- iii) Native method interface

7) How does Java achieve platform independence through the JVM?

7)
 Ans

1) Java is platform independent through JVM because we first compile it and then run the program.

ii) Once we compile and get .class file name this class file is generated.

iii) Class or compiled file can be run on any different OS irrespective of JRE.

iv) JRE is required then we can run the file i.e byte code in any system.

Page No.
 Date : / / 20

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

Q.8)

Ans

- i) Class loader is the component which is used to load the .class file to the JVM.
- ii) Either class file are loaded from .jar or folder from exe Folder in library.
- iii) Class loader have 4 main components
 - ↳ Bootstrap Class loader
 - ↳ Extension Class loader
 - ↳ System Class loader
 - ↳ ~~Class~~ Custom Class loader
- iv) Garbage collector remove the memory used by applications.
- v) It prevents memory leakage and optimize the memory. It uses various algorithms to remove the data.

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

9)

Ans

1) Modifiers are used to control the visibility of the members class (enum) interface.

11) 4 access modifiers in Java.

- ① public
- ② private
- ③ protected
- ④ default (package level private)

111) (A = Accessible)

Access modifier	Same Class	Same Package		Diff. Package	
		Sub Class	Non Sub Class	Sub Class	Non Sub Class
public	A	A	A	A	A
private	A	-	-	-	-
protected	A	A	A	A	-
default	A	A	A	-	-

10) What is the difference between public, protected, and default access modifiers?

10)
Ans

public - public access modifiers are used in all classes, subclass & packages.

private - private means access only within the class.

protected - It is used in same class, sub-class, non-sub class & package anywhere within the sub class.

default - package within protected of only one class.

11) Can you override a method with a different access modifier in a subclass? For example, can a protected method in a superclass be overridden with a private method in a subclass? Explain.

11)
Ans

- i) No, we cannot override a method with different ~~etc~~ restrictive access modifier in a subclass
- ii) We can override protected or public method in subclass
- iii) We can't override private also since its private and can't be access to other classes.

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

12)
Ans

The diffⁿ between protected & default access modifier are

- Protected : Members accessible within same package & subclass even though they are in different packages.
- Default : Accessible ~~at~~ only within different packages. but not outside.

13) Is it possible to make a class private in Java? If yes, where can it be done, and what are the limitations?

13)
Ans Yes It is possible to make a class ~~in~~ private in Java but only can be done with the help of nested class.

eg

```
public class OuterClass {  
    private class InnerClass {  
        //code  
    }  
}
```

14) Can a top-level class in Java be declared as protected or private? Why or why not?

14)
Ans

- i) No, top class cannot be declared as protected.
- ii) Top class can be public or package level but not protected or private since then we will not be able to access it.

Page No. _____
Date : ____ / ____ / 20__

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

15)

Ans.

Private modifier - The access modifier restricts only within variable or method in the class which is declared. No other class with same package can access that private member.

eg

```
class A {  
    private int Var = 10;  
  
    private void pmMethod() {  
        SOP ("method in class A");  
    }  
}  
  
class B {  
    public void accessPrivateMembers() {  
        class A obj = new class A();  
        SOP (obj.privateVariable);  
        obj.privateMethod();  
    }  
}
```

// Here class B cannot access either the variable or method from class A due to compilation errors.

16) Explain the concept of "package-private" or "default" access. How does it affect the visibility of class members?

