CDAC Mumbai PG-DAC AUGUST 24

Assignment No- 3

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

1) Explain the components of the JDK.

-> It is an software development kit and in java we call it as Java Development kit . It contains JVM and JRE and many things like libraries , java interpreter ,loader , basically it provides the development environment so that the developer can easily make the applications.

2) Differentiate between JDK, JVM, and JRE.

JVM -> It is known as java virtual machine and it is used to run the bytecode of program , it can also run the c and cpp bytecode if they are compiled into the java bytecode.

JRE -> It is an runtime environment and in javs we call it as JRE (Java Runtime Environment) where it contains all the libraries and files that JVM used to execute the code.

JDK -> Jave development kit contains JVM , JRE in in and also the rt.jar file which really important to execute the code and It also contains the development tools to develop the software.

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

-> The role of JVM is to provide the run time environment to JAVA bytecode . And the process is that it fist load the code the verify the code then provide the runtime to execute the code. It is an platform dependent so we need different JVM for different OS .

4) Explain the memory management system of the JVM.

-> Memory management in JVM happens on free some space for objects by removing the garbage present in it . When we create the new object it goes into the heap and heap contains two type of allocation nursery (Young allocation) and old allocation and if the young allocation is full and we want to assign the new object then the old assignments of objects are forwarded to old allocation and free some space in young allocation and if the old allocation also get full then the garbage present in it is going to collect to garbage collector to free the space on it.

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

-> JIT compiler is used to covert the byte code into native java code and when jvm first runs the java byte code it acts as interpreter and when it runs the code again now because it was compile into the native code so now it works as a compiler to just call the method.

Byte code is an intermediate code generated by the compiler and it is very important because it makes java an independent language by using byte code (.class) we can run it any OS where the JVM is present.

6) Describe the architecture of the JVM.

-> JVM firstly load the code with the help of loader and the it allocates the memory by memory management and then code goes to the execution engine and then compiles the code.

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

We need different JVM for different OS that’s why it is platform dependent and Java achieve is independent because after compilation of the codes we get the byte code and we can run byte code in any platform where the JVM is installed so it is platform independent.

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

Java ClassLoader is an abstract class. It belongs to a **java.lang** package. It loads classes from different resources. Java ClassLoader is used to load the classes at run time.

The objects present in the old allocation in head and which are unused the garbage collector will destroy these object to free some space for new objects.

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

-> Public, Private , Protected and default . Public means we can access it any where.

Private means we cannot inherit it any were and Protected means we can access it in all subclasses rather it is on the packages or not and default means we cannot access out side of the package.

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

Public means we can access it any where. Protected means we can access it in all subclasses rather it is on the packages or not and default means we cannot access out side of the package.

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.

Yes we can override a method with a different access modifier but the condition is that we have to define the weaker modifier to the subclass or at least same as compare to parent class.

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

The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default. Protected: The access level of a protected modifier is within the package and outside the package through child class.

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

Yes you can declare a class with private access specifier. But not in main class. You can declare private in inner classes only.

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

No, we cannot declare a top-level class as private or protected. It can be either public or default .

Because a class with private or protected mofdifier is completely use less.

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?

Private member cannot be accessed to another class but we can set the getter and setter to set the values and get the values.

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

If the members in a class is set as default the it can only access within the package and not on outside of the packages.