

Q1 Explain the components of the JDK.

Ans:-

① Java Development tools:-

① Compiler (javac) :-

Javac compiler is used for  
to compile the source code (.java)  
to byte code (.class).

The generated byte code is a platform  
independent.

② Documentation Java API:-

In API, having a pre-defined  
packages , all classes with a specific  
meaning.

Ex: Java.lang (packages)

having @ interfaces, classes, enums, etc.

③ Java Execution Environment:-

Ex: JVM

Java Virtual Machine is a runtime  
polymorphism that executes the byte  
code (.class file)

Main method call by JVM which  
is present in source code.  
JVM is a part of JRE.

#### ④ Java Runtime Environment:-

JRE is a part of JDK that contains Java class library.

#### ② Differentiate betw. JDK, JVM & JRE

Ans:-

JDK :- It is a complete toolkit for developing java source code, including JRE and JVM.

JRE :- JRE is a part of JDK it contains the java class library.

It contains everything needed to run java program.

JVM: JVM provides a runtime environment in which java bytecode can be executed.

JVM are available for many hardware and software platform. JVM is a platform dependent.

③ What is the role of the JVM in Java? How does the JVM executes Java code?

Ans:-

① JVM executes a bytecode is responsible for converting a bytecode into native code.

② JVM loads a .class file. JVM performs different different task to execute a file.  
ex:- ① loads code

② verify code

③ executes code

④ provides runtime environment.

④ Explain the Memory Management system of the JVM?

Ans:-

In JVM having a five different memory area are

① Method area = stores static fields or methods.

② Heap = stores a instance of a class

③ Stack = stores a reference variable.

④ PC register = stores the address of current instruction.

⑤ Native method = creation of new threads, memory allocate for each thread.

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

Ans:-

① JIT stands for Just In Time Compiler. If the method is executed frequently then JIT maintains a record that how many times execute method.

Then next time it will not interpret the method and compile the method.

② Bytecode is converted from source file. Bytecode means binary files 0's & 1's.

③ Because computer understand only binary language.

- Q) Describes the architecture of ~~Java~~ JVM?

Ans:- ① In JVM having a 3 stages.

a) class loader      b) memory area  
c) execution

a) class loader loads .class file stores its bytecode in memory.

b) verifier verifies verify the instruction that are in the memory, & JVM interpreter executes those instruction.

7 How does Java achieve platform independence through the JVM?

Ans

- ① By compiling the source code it generates a 'Byte code', which is executed on any system.
- ② Byte code is platform independent.
- ③ Install JVM & runs Byte code.

8 What is the significance of the class loader in Java & what is the process of garbage collection in Java?

Ans:

- ① Class loader is used to load the classes at run time. classes loaded according to JVM.
- ② Garbage collector automatically frees up memory space.

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

Ans:- four access modifiers are

- ① private
- ② package level private
- ③ protected
- ④ public

① Private field & methods are access within same class.

② package level private access in same class package.

③ Protected & ~~private~~ public access in same & different package.

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

Ans:-

- ① Public methods are accessible in same package and different package.
- ② Protected methods are accessible in same package and in different package is only in sub class.
- ③ Default access Only in same package.

(ii) Can you override a method with different access modifier in a subclass? For ex:- can a protected method in a superclass be overridden with a private method in a subclass? Explain.

Ans:-

No, we can't reduce the visibility of a ~~method~~ overriding method.

If parent method is protected then overwritten method must be protected or public.

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

Ans:-  
protected methods are accessible in same package and in different class only in sub class.

Default access only in same package.

Q) Is it possible to make class private in Java? If yes, where it can be done and what are the limitation.

Ans:-  
① In Java, we can't make a top-level class as a private.

② But we can make a class private if it is ~~is~~ within class & private class are accessible only in same body.

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

Ans:- We can declare a top-level class as a protected but not as a private because protected can access within a same package.

private not accessible within a same package.

- Q15 Explain what happen if you declare a variable or method as a private in a class and try to access it from another class within a same package.
- Q16 If we declare method or variable as a private, we can access that method & variable within same class. We can't access private method or variable in different class.
- Q17 Explain the concept of "package-private" or "default" access. How does it affect the visibility of class member?
- Q18 We ~~access~~ can access the default variable or methods only in same package. We can't use default variable or method in different package.