**1) What are JDK, JVM, and JRE?**

**1. JDK** (Java Development Kit) is a Kit that provides the environment to **develop and execute (run)** the Java program. JDK is a kit( or package) that includes two things

* Development Tools(to provide an environment to develop your java programs)
* JOE

**2.**[**JVM** (**Java Virtual Machine)**](https://www.geeksforgeeks.org/jvm-works-jvm-architecture/)is a very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for executing the java program line by line, hence it is also known as an [**i*nterpreter***](https://www.geeksforgeeks.org/compiler-vs-interpreter-2/)**.**

**JRE** (Java Runtime Environment) is an installation package that provides an environment to **only run(not develop)** the java program(or application)onto your machine. JRE is only used by those who only want to run Java programs that are end-users of your system.

**2) What is the OOPs concept in Java with real-time examples?**

Oops comes with four main features encapsulation, abstraction, inheritance, and polymorphism. When we write a program using these features. The main goal of the OOPs concept in java programming is that everything you want to do, do through objects. Using the apps we can create the applications. And Topps is related to our real-time life for example in apps the concept is called inheritance. Inheritance means we can create a class and we can call that class a parent class and parent means children also come and we can create the child classes. And these child classes can have all properties of the parent class just like our real life

**3) Explain Polymorphism, Abstraction, and Interface in Detail.**

as the name suggests that polymorphism means many forms.

There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism.

We can perform polymorphism in java by method overloading and method overriding.

If you overload a method in Java, it is an example of compile-time polymorphism.

and if we override a method it is an example of runtime polymorphism.

Abstraction: we can create the abstraction class in java and we can call that abstract class as

A partially implemented class.because it can contain the abstract methods which just can have

A method signature and do not have any method body. and as well as we can create the normal methods

inside an abstract class. and the condition to that abstract class is that it should have a child class

and in that class, we have to override all abstract methods in their child classes.

**4) Overriding vs Overloading.**

1) Overriding implements Runtime Polymorphism whereas Overloading implements Compile time polymorphism.

2) The method Overriding occurs between superclass and subclass whereas Overloading occurs between the methods in the same class.

In case of overloading, we can create the same methods in the class and the main thing Is that we can not create the two methods with the same parameters. The parameters should be different.

And overloading happens in different classes and for that the methods in the parent class should be there in the child class with the same parameters and the first preference will be given to the overridden methods.

**5) Garbage collector how it works in java**

In Java, the garbage collector is the process of managing memory, automatically. It finds the unused objects (that are no longer used by the program) and deletes or removes them to free up the memory. The garbage collection mechanism uses several GC algorithms.

When a program executes in Java, it uses memory in different ways. The heap is a part of the memory where objects live. All the garbage collection makes sure that the heap has as much free space as possible. The function of the garbage collector is to find and delete objects that cannot be reached.