

## **Explain polymorphism.**

Polymorphism in Java is a concept by which we can perform a single action in different ways. We can perform polymorphism in java by method overloading and method overriding.

## **What is overloading?**

Method overloading is a feature of Java in which a class has more than one method of the same name and their parameters are different

## **What is overriding?**

Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its superclasses or parent classes

## **What does the final mean in this method: `public void doSomething(final Car aCar){}`**

The final means that the object Car is immutable and cannot be changed in the method doSomething

## **Suppose in question 4, the Car class has a method `setColor(Color color){...}`, inside doSomething method, Can we call `aCar.setColor(red);`?**

Yes, we can call the method setColor as we are not reassigning aCar

## **Can we declare a static variable inside a method?**

You can't declare a static variable inside a method, static means that it's a variable/method of a class, it belongs to the whole class but not to one of its certain objects.

## **What is the difference between interface and abstract class?**

An interface is an empty shell. There are only the signatures of the methods, which implies that the methods do not have a body. The interface can't do anything. It's just a pattern.

Abstract classes can have constants, members, method stubs (methods without a body) and defined methods, whereas interfaces can only have constants and methods stubs

## **Can an abstract class be defined without any abstract methods?**

Yes. Declaring a class abstract only means that you don't allow it to be instantiated on its own. Declaring a method abstract means that subclasses have to provide an implementation for that method.

## **Since there is no way to create an object of abstract class, what's the point of constructors of abstract class?**

We need a constructor to initialize the non-abstract methods and instance variables

**What is a native method?**

Native methods allow you to use code from other languages such as C or C++ in your java code. You use them when java doesn't provide the functionality that you need.

**What is the marker interface?**

A marker interface is an interface that has no methods or constants inside it. It provides run-time type information about objects, so the compiler and JVM have additional information about the object.

**Why to override equals and hashCode methods?**

If two objects are considered equal, their hashcodes must also be equal otherwise you'd never be able to find the object since the default hashCode method in class Object virtually always comes up with a unique number for each object. If two objects are equal, their hashcodes must be equal as well.

**What's the difference between int and Integer?**

In Java, int is a primitive data type while Integer is a Wrapper class. ... Integer is a class and thus it can call various in-built methods defined in the class. Variables of type Integer store references to Integer objects

**What is serialization?**

Serialization is simply turning an existing object into a byte array. This byte array represents the class of the object, the version of the object, and the internal state of the object. This byte array can then be used between JVM's running the same code to transmit/read the object.