**Theory 2**

**Question 1: What is Constructor to do? What is the default constructor?**

Constructor to initialize the object.

The default constructor is a constructor with no parameters and it is used to initialize the default values ​​for the object.

**Question 2: What is a class and an object?**

A class is a template describing common properties and behaviors of a group of objects.

An object is a specific instance of the class. Objects have specific characteristics and behaviors.

**Question 3: What is Overloading?** Many methods in a class have the same name but different input parameters.

**Question 4: What is Overriding?** A child class that redefines the method of the parent class. (The subclass's method has the same name, same input parameters as the parent class)

**Question 5: What does the keyword static in Java mean?**

Static is used to declare a variable or method of a class. Accessing these variables or methods can use the class name, not the object's initialization.

Static methods cannot be overridden.

**Question 6: What is static block?**To initialize static data members**.** It is executed before the main method

**Question 7: What is final variable?** A variable whose value cannot be changed

**Question 8: What is the final method?** The method cannot be overridden.

**Question 9: What is final class?** A class that cannot be inherited

**Question 10: What is final blank variable?** The final variable is not initialized.

**Question 11: If I want to assign value to final blank variable, what should I do?**

If the variable is not static, the value must be assigned in the constructor

If the variable is static, you must assign the value in static block

**Question 12: What are the properties of OOP?** Abstract, inheritance, polymorphism, encapsulation

**Question 13: What is abstraction?**

The abstraction is to hide implementation details and only show the features to the user. This property allows the elimination of complex properties of the object, just give the necessary properties in programming to focus on the object's core.

**Question 14: What is an Abstract class?** A class that cannot be specified.

**Question 15: What is Abstract method?** A method without an implementation

**Question 16: Where should the abstract method be located?** Stay in the abstract class.

**Question 17: What are special properties of the interface?** Must be a constant

**Question 18: What is special about the method of the interface?**No body**.** The public method

**Question 19: Can the Interface be final?** Are not. Because you want to use the method of the interface, there must be a class that implements it and uses it through that class.

**Question 20: Is the abstract class final?** Are not. Because it is necessary to have other classes inheriting in order to have different instances of the abstract class.

**Question 21: What is encapsulation?** Specifies that the object's states do not allow code that is outside of the scope to change state or visibility.

**Question 22: Where is the encapsulation shown?** Access modifier

**Question 23: What are the types of Access modifiers and what are their scope of use?**

Public: accessible anywhere

Protected: accessible by classes in the same package or subclasses

Default: accessible by classes in the same package

Private: can be accessed in the same class

**Question 24: What is inheritance?** Is the ability to build a new class (child) based on an existing class (parent).

**Question 25: What class is the father of all classes?** Object

**Question 26: If we want to refer directly to the objects of the parent class, which keyword do we use?** Super

**Question 27: What is the difference between this () and super ()?**

This () calls to the constructor in the current class and super () calls to the constructor in the parent class

**Question 28: What is polymorphism?** When an action is performed in different ways.

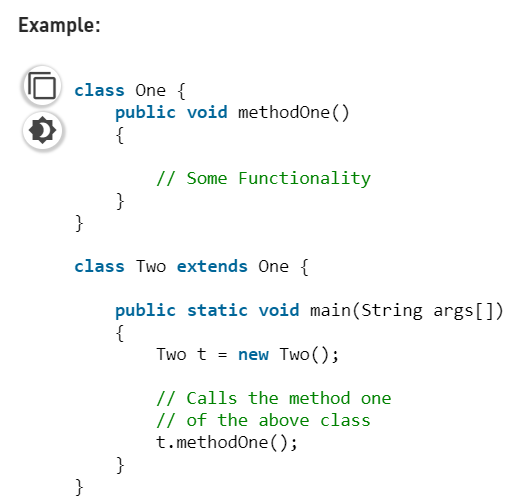
**Question 29: How can polymorphism be possible?** Use overloading or overriding

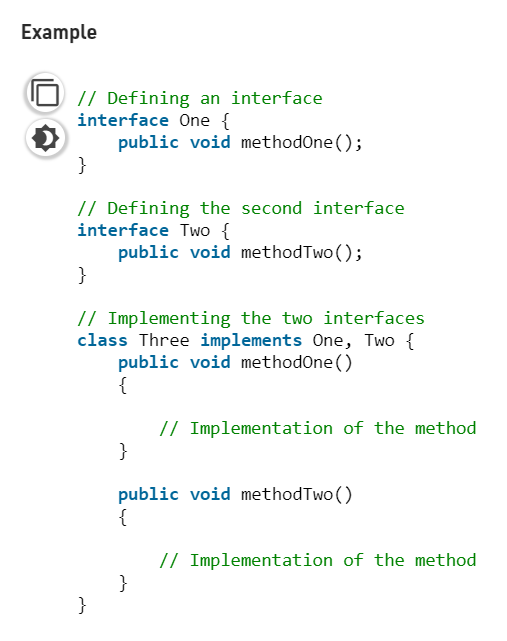
**Question 30: Abstract and Interface**

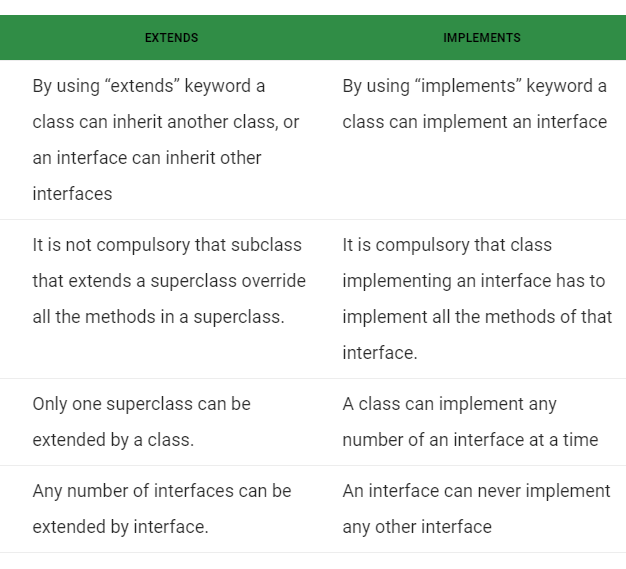
Abstract class: is a parent class for all classes with the same nature. The essence here is understood as the type, type, and duties of the class. Two classes that implement the same interface can be completely different in nature. Understand simply as a child (child class) can only be a son of a father, has the same personality (abstract class) it.

Interface: is a function that you can add to any class. The word function is not synonymous with method (or function). Interface can consist of multiple functions / methods and all of them serve the same function.

**Question 31: Extends and Implements**

**Extends:** In Java, the extends keyword is used to indicate that the [class](https://www.geeksforgeeks.org/classes-objects-Java/) which is being defined is derived from the base class using inheritance. So basically, extends keyword is used to extend the functionality of the parent class to the subclass. In Java, multiple inheritances are not allowed due to ambiguity. Therefore, a class can extend only one class to avoid ambiguity.



**Implements:** In Java, the im plements keyword is used to implement an [interface](https://www.geeksforgeeks.org/interfaces-in-Java/). An interface is a special type of class which implements a complete abstraction and only contains [abstract methods](https://www.geeksforgeeks.org/abstract-methods-in-Java-with-examples/). To access the interface methods, the interface must be “implemented” by another class with the implements keyword and the methods need to be implemented in the class which is inheriting the properties of the interface. Since an interface is not having the implementation of the methods, a class can implement any number of interfaces at a time.