***OOP summary***

What are the OOP concepts in Java do you know?

1. Encapsulation.

2. Inheritance.

3. Abstraction.

4. Polymorphism.

What is an encapsulation in java and how do we achieve it?

* It is data protection(hiding) mechanism. We set properties as private and we provide getters and setters.

What's POJO?

* Plain Old Java Object. Usually encapsulated objected referred as POJO.

What's inheritance in java?

* It is a process where one class can inherit visible properties from another class.

What's super class?

* It is parent class in inheritance.

What's the class which every class will inherit from?

* java.lang.Object

Why?

* To give generic behaviors for every object. For example, equals, hashCode, toString.

How many classes we can extend at once?

* Just one.

Why java doesn't allow multiple inheritance?

* because of diamond problem.

What is the difference between method overriding and method overloading?

* overloading when we have methods with same name but different arguments withing the same class.
* we can override parent class or interface methods.

Rules:

* same method name
* same arguments
* same access modifier or child class method can be more visible
* return type should be same or covariant.
* if parent method declares exception, the child method can declare same or smaller exception. (or none).

What does 'this' and 'super' keyword mean in java?

* 'super' keyword is used to access the parent class properties and methods
* 'this' is used to access current object properties and methods

What's constructor in java?

* It is special method to create an object.

What's default constructor?

* If we don't define any constructors by ourselves java will provide one default(empty) constructor.

What's abstraction in java?

* When we focus on what object does instead of how it does it.

How do we achieve abstraction?

* By using abstract class and interface.

What's the difference between abstract class and interface?

* We cannot create object from both of them. They always used with concrete classes.
* abstract class can have both abstract and non-abstract methods and interface can have only abstract methods.
* abstract class can have instance variables and interface can have only public static final.
* we can extend only one abstract class but we can implement many interfaces.

What are two methods that can have bodies in the interface?

1. default method.

2. static method.

What is a concrete class in abstraction?

* It regular class that extends/implements abstract class/interface
* it must implement (Override) all abstract methods.

How to prevent method to be overridden?

* by making it final

How to prevent class to be extended?

* by making it final as well

What's polymorphism in java?

* It is ability of an object to take many different forms.

Map<String, String> map = new HashMap<>();

Map<String, String> map2 = new LinkedHashMap<>();

Map<String, String> map -> reference type

Reference type decides what methods and properties are available.

Runtime polymorphism is when java thinks that method will be called from reference type but the actual object overridden this method. Then the overridden method will be called. Java will get to know about overriding during the runtime only, that's why it is called runtime polymorphism.