1. **What is difference in Overriding and Overloading?**

# Overriding implements Runtime Polymorphism whereas Overloading implements Compile time polymorphism.

# The method Overriding occurs between superclass and subclass. Overloading occurs between the methods in the same class.

# Overriding methods have the same signature i.e. same name and method arguments. Overloaded method names are the same but the parameters are different.

# With Overloading, the method to call is determined at the compile-time. With overriding, the method call is determined at the runtime based on the object type.

# If overriding breaks, it can cause serious issues in our program because the effect will be visible at runtime. Whereas if overloading breaks, the compile-time error will come and it’s easy to fix.

1. **If both the functions are static, can we override?**

* No, we cannot override two methods if they differ only by static keyword.

1. **Difference between Final, Finally and Finalize .**

* Java Final keyword is a non-access specifier that is used to restrict a class, variable, and method
* The Finally keyword is used to create a block of code that follows a try block/catch block
* Finalize is the method in Java which is used to perform clean up processing just before an object is garbage collected

1. **What are maker Interfaces?**

* An [interface](https://www.javatpoint.com/interface-in-java) that does not contain methods, fields, and constants is known as **marker interface,** it indicates a signal or command to the JVM.
* It delivers the run-time type information about an object.
* It is the reason that the [JVM](https://www.javatpoint.com/jvm-java-virtual-machine) and compiler have additional information about an object.
* The **Serializable** and **Cloneable** interfaces are the example of marker interface.

1. **What is Singleton class?**

* A singleton class is a class that can have only one object that is one instance of the class at a time.

1. **What is the underlying Data structure for Array list and Linked list?**

* Array List uses the Array data structure, and LinkedList uses the Doubly Linked List data structure.

1. **What is hash collision?**

* a hash code collision in a HashMap, is a situation where two or more key objects produce the same final hash value and hence point to the same bucket location or array index.

1. **What are lambda Expressions?**

* A lambda expression is a short block of code which takes in parameters and returns a value. Lambda expressions are similar to methods, but they do not need a name and they can be implemented right in the body of a method.

1. **What is functional interface?**

* A functional interface is an interface that contains only one abstract method.

**10.What are optional classes in Java 8?**

**-**  optional class is a class which is used to avoid null checks and null pointer exception.

**11.Whats is factory design pattern and Adapter and difference between them?**

**-** A Factory Pattern is an interface or abstract class for creating an object but let the subclasses decide which class to instantiate.

- Adapter Design pattern works as a bridge between two incompatible interfaces. This type of design pattern comes under structural pattern as this pattern combines the capability of two independent interfaces.

- Factory Pattern does the operation on the created objects (the factory class done the job after creation), whereas Adapter pattern is used after the application components are designed so that we can use them without modifying the source code.

**12.What are Solid principles?**

-These principles establish practices that lend to developing software with considerations for maintaining and extending as the project grows.

-Adopting these practices can also contribute to avoiding code smells, refactoring code, and Agile or Adaptive software development.

-It’s a mnemonic acronym for the following five design principles:

* Single Responsibility Principle
* Open/Closed Principle
* Liskov Substitution Principle
* Interface Segregation Principle
* Dependency Inversion

**13. What does @component and @scan do in Sprinboot annaotation**

- @Component is an annotation that allows Spring to automatically detect our custom beans.

In other words, without having to write any explicit code, Spring will Scan our application for classes annotated with @Component. Instantiate them and inject any specified dependencies into them.

With Spring, we use the @ComponentScan annotation along with the @Configuration annotation to specify the packages that we want to be scanned.

@ComponentScan without arguments tells Spring to scan the current package and all of its sub-packages.

**14. What is the Difference between JPA , Hibernate and ORM?**

- Java Persistence API (JPA) defines the management of relational data in the Java applications.

* Hibernate is a tool which is used to save the state of Java object into the database. It is just a specification. Various ORM tools implement it for data persistence.
* Object-Relational Mapping (ORM) is the process of converting Java objects to database tables. In other words, this allows us to interact with a relational database without any SQL.

**15.What is indexing in Database?**

- Indexing is a way to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed.

It is a data structure technique which is used to quickly locate and access the data in a database.

**16.What are 4 types of Drivers in JDBC?**

- There are five types of JDBC drivers in use:

Type 1: JDBC-ODBC bridge

Type 2: partial Java driver

Type 3: pure Java driver for database middleware

Type 4: pure Java driver for direct-to-database

Type 5: highly-functional drivers with superior performance

**17.What is difference between @component and @bean and @service?**

-@bean - It is used to explicitly declare a single bean, rather than letting Spring do it automatically.

-@component - If any class is annotated with @Component it will be automatically detect by using class path scan.

-@Service is a stereotype for the service layer. It annotates classes at the service layer.

**18. What is difference between SOAP and restful services technically?**

- SOAP stands for Simple Object Access Protocol

* REST stands for Representational State Transfer.
* REST is a set of guidelines that offers flexible implementation, whereas SOAP is a protocol with specific requirements like XML messaging.

**19 . Difference in POST and PUT ?**

* PUT method is call when you have to modify a single resource, which is already a part of resource collection.
* POST method is call when you have to add a child resource under resources collection.

**20. Is react uni directional or Bi directional ?**

- React, a Javascript library, uses unidirectional data flow and doesn't support bi-directional binding.