Java Interview Questions

----------------------------------

1. What is Oops and its features.
2. What is Interface where we are implementing it.
3. What is Abstract class where we are implementing it.
4. What is the difference between Interface and Abstract class.
5. What is Polymorphism. What is static and dynamin binding polymorphism.
6. What is JVM. how it works.
7. What is Class loader. Describe it.
8. Why String is immutable.
9. What is the difference between StringBuffer and StringBuilder.
10. What is == and hash ().
11. What is Exception Hierarchy.
12. What is Exception. How to handle Exception in java.
13. What is ClassNotFoundException, NoClassDefFoundException
14. What is Garbage collector. How it works.
15. What is Multithreading.
16. How many ways to create a Thread.
17. What is synchronization.
18. Describe Thread life cycle.
19. What is Deadlock, how to prevent it.
20. Collection vs Collections
21. ArrayList vs LinkedList.
22. HashSet vs LinkedHashSet
23. What is HashMap ,how it work internally.
24. HashMap vs ConcurrentHashMap
25. What is FailFast and FailSafe.

Java 8 Interview Questions

---------------------------------

1. What is Functional Interface
2. What is StreamAPI
3. What is Optional Class.
4. What are the difference between map() vs flatmap()
5. What are the method references? What is the use of them?
6. What are intermediate and terminal operations?
7. What are the differences between collections and streams?

Spring Interview Questions

----------------------------------

1. What is Dependency Injection.
2. What is IoC.
3. What is Spring Bean.
4. What is Autowire. And its implementation.
5. What are the ways to create a Bean in Spring.
6. Stereotype Annotation. (@controller,@component,@service,@repository)
7. What is @profile
8. What is AOP.

SpringBoot Interview Questions

----------------------------------------

1. What is SpringBoot. Advantages of SpringBoot.
2. What is SpringBoot actuator. How it works.
3. What is AutoConfiguration in SpringBoot.
4. How to override the autoconfiguration or custom configuration in springboot.
5. What is application.properties or application.yml file.
6. What is ORM,
7. What is JPA.
8. What is the namedQuery vs nativeQuery
9. What is RestTemplate and WebClient.
10. What is ResponseBody and RequestBody in Rest.
11. What is @RequestMapping
12. What is @RequstParam
13. What is @PathVariable
14. What is RequestBody and ResponseBody
15. Http methods
16. Status code
17. What is Rest. And its methods
18. What is POST Vs PUT Vs Patch
19. What is URI and how it works in REST.

Microservice Interview Questions

------------------------------------------

1. What is Microservice. What are the benefits of Microservice Architecture.
2. What are the merits and demerits of Microservice Architecture.
3. What is Service Registry. And its implementation
4. What is Circuit Breaker and its implementation?
5. What is API Gateway and its implementation?
6. What is load balancer and its implementation?
7. What is fault tolerance.
8. How to handle multiple transactions in multiple microservice.

Kafka Interview Questions

---------------------------------

1. What is Kafka.
2. What is Zookeeper.
3. What is Topics
4. What is Consumer
5. What is Producer
6. What is broker.
7. What is partition
8. What is repica

**1. What is Oops and its features.**

Ans. Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

OOPs has several advantages over procedural programming:

* OOP is faster and easier to execute.
* OOP provides a clear structure for the programs.
* OOP helps to keep the Java code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug.

**2. What is Interface where we are implementing it.**

Ans. Interfaces in java are very much like abstract classes but interfaces contain only abstract methods. Abstract classes may contain both abstract methods as well as concrete methods. But interfaces must contain only abstract methods. Concrete methods are not allowed in interfaces. Therefore, Interfaces show 100% abstractness.

* Interfaces are declared with keyword ‘interface‘ and interfaces are implemented by the class using ‘implements‘ keyword.
* Interfaces should contain only abstract methods. Interfaces should not contain a single concrete method.
* By default, every field of an interface is public, static and final.
* Any class in java cannot extend more than one class. But class can implement more than one interface. This is how multiple inheritance is implemented in java.

**3. What is Abstract class where we are implementing it.**

Ans.Abstraction is used to separate ideas from their implementation. Abstraction in java is used to define only ideas in one class so that the idea can be implemented by its sub classes according to their requirements.

* Abstract classes and abstract methods are declared using ‘abstract‘ keyword. We can’t create objects to those classes which are declared as abstract. But, we can create objects to sub classes of abstract class, provided they must implement abstract methods.
* The methods which are not implemented or which don’t have definitions must be declared with ‘abstract’ keyword and the class which contains it must be also declared as abstract.
* You can’t create objects to abstract class even though it does not contain any abstract methods.
* Abstract methods can not be private. Because abstract methods must be implemented somehow in the sub classes. If you declare them as private, then you can’t use them outside the class.

**4. What is the difference between Interface and Abstract class.**

|  |  |
| --- | --- |
| Abstract class | Interface |
| 1.Abstract class can have abstract & not abstract methods.  2. Abstract class doesn’t support multiple inheritance.  3. Abstract class **can have final, non-final, static and non-static variables**.  4. Abstract class **can provide the implementation of interface.**  5. The **abstract keyword** is used to declare abstract class.  6. An **abstract class** can extend another Java class and implement multiple Java interfaces.  7. A Java **abstract class** can have class members like private, protected, etc. | 1. Interface can have only abstract methods. It can have multiple default and static methods.  2. Interface **supports multiple inheritance**.  3. Interface has **only static and final variables.**  4. Interface **can't provide the implementation of abstract class**.  5. The **interface keyword** is used to declare interface.  6. An **interface** can extend another Java interface only.  7. Members of a Java interface are public by default. |

**5. What is Polymorphism. What is static and binding polymorphism.**

Ans. Polymorphism in java refers to any entity whether it is an operator or a constructor or any method which takes many forms or can be used for multiple tasks either while compiling or while running a java program.

**Static Polymorphism:** Any entity which shows polymorphism during compile time is called static polymorphism. Operator Overloading, Constructor Overloading and method overloading are best examples of static polymorphism. Because they show polymorphism during compilation. In static polymorphism, the object used is determined during compilation itself. So, it is called static binding or Early Binding.

**Dynamic Polymorphism:** Any entity which shows polymorphism during run time is called dynamic polymorphism. Method Overriding is the best example of dynamic polymorphism. It is also called dynamic binding or late binding, because type of the object used will be determined at run time only.

Q.

**6.**

**What is JVM. how it works.**

**7. What is Class loader. Describe it.**

**8. Why String is immutable.**

**9. What is the difference between StringBuffer and StringBuilder.**

**10. What is == and hash ().**

**11. What is Exception Hierarchy.**

**12. What is Exception. How to handle Exception in java.**

**13. What is ClassNotFoundException, NoClassDefFoundException**

**14. What is Garbage collector. How it works.**

**15. What is Multithreading.**

**16. How many ways to create a Thread.**

**17. What is synchronization.**

**18. Describe Thread life cycle.**

**19. What is Deadlock, how to prevent it.**

**20. Collection vs Collections**

**21. ArrayList vs LinkedList.**

**22. HashSet vs LinkedHashSet**

**23. What is HashMap ,how it work internally.**

**24. HashMap vs ConcurrentHashMap**

**25. What is FailFast and FailSafe.**