

1. What is an interface in Java?

An interface in Java is a blueprint for a class that specifies a set of methods (without any implementation or abstract methods) that a class must implement. It is used to achieve abstraction and multiple inheritance in Java.

```
interface Animal {  
    void eat(); // Abstract method  
    void sleep(); // Abstract method  
}  
  
class Dog implements Animal {  
    public void eat() {  
        System.out.println("Dog eats.");  
    }  
  
    public void sleep() {  
        System.out.println("Dog sleeps.");  
    }  
}
```

2. Which modifiers are allowed for methods in an Interface? Explain with an example

Before Java 8: Methods in an interface could only be public and abstract.
From Java 8:

- Default methods: Can have an implementation using the default keyword.
- Static methods: Can have an implementation and belong to the interface itself.

From Java 9:

- Private methods: Can be used for common functionality shared by other methods within the interface.

3. What is the use of interface in Java? Or, why do we use an interface in Java?

We use interface because -

Achieve Abstraction: It hides implementation details and only shows functionality.

Multiple Inheritance: A class can implement multiple interfaces, unlike extending a single class.

Polymorphism: Interfaces allow different classes to be treated uniformly based on their behavior.

Separation of Concerns: Interfaces help separate the definition of behavior from its implementation.

Easy to extend : Software can easily be extended with new features and hence software becomes scalable.

Decoupling : It promotes decoupling in software design. Hence, it avoids tightly bound classes in the software design.

4. What is the difference between abstract class and interface in Java?

Abstraction :

- It can have abstract and as well as non-abstract concrete methods.
- It can have instance variables and constants.
- A class can extend only one abstract class
- It can have public, protected, or private methods.

Interfaces

- Only abstract methods (default, static, and private methods later on added in Java8+).
- Only public, static, and final variables are allowed in interfaces
- A class can implement multiple interfaces
- All methods are public by default in interfaces