1. What is an interface in Java?
Ans: An interface is a block of code similar to a class but not the same and is used to achieve full or 100% abstraction. It contains only abstract methods and constant variables. All variables and methods in it are public by default. No object can be created for an interface.

2. Which modifiers are allowed for methods in an Interface? Explain with an example.
Ans:Only public and abstract modifiers are allowed for methods in an Interface.
This is because:
i. If we implement the method in the interface it no longer remains abstract as abstract method menas that the implementation is not present at the place where method is written.
ii. If we give any other access modifier other than public then it can't be implemented in any of the Child classes of the interface. For implementing th method in the interface somewhere else, it is mandatory to make it public.

3. What is the use of interface in Java? Or, why do we use an interface in Java?
Ans:There are multiple reasons for using an interface.The use are as follows:
i. 100% abstraction is possible by using an interface.
ii. Using interfaces, we can achieve the Multiple Inheritance in Java.
iii. Since interface acts as a contract between a client and service provider, from client point of view, we can defined the set of software services that is expected by them and from Service Provider point of view, interface defines the set of software services that they are offering. In this way, prograanners can customize features of a software differently.
iv. Using interfaces are the best way to expose the APIs(services) of the current project to another project.

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

Abstract Class	**Interface**
1. It has partial implementation, i.e, some methids may have their implementation inside the class, some may not, i.e, it has abstract as well as non-abstract methods.	1. None of the methods in the interface have any implementation, i.e, all methods are abstract. But since Java 8, it can have default and static methods also.
2.There are no restrictions on access specifiers of the methods in it	2. Cannot use any access specifier other than public for the methods in interface. Since Java 8, we can use default and static methods.
3. Every abstract class variable need not be public static final	3. Interface has only public static final variables irrespective of whether we declare it explicitly or not.
4. A constructor can be defined for an abstract class.	4. No constructor can be defined for an interface.

5. Multiple inheritance is not allowed for abstract classes.	5. Mutlpce inheritance is allowed for interfaces. Ex: implements A, B
6. extends keyword is used to implement the abstract methods in the Child class.	6. implements keyword is used to implement the abstract methods in a separate class.
7. Members of an abstract class can be public, private, protected etc.	7. All members of an interface are public by default..
8. An Abstract class can provide the implementation of an interface.	8. An Interface cannot provide the implementation of an abstract class.
9. An abstract class can extend another Java class and can implement multiple interfaces.	9. An interface can extend another interface only.

10. Example:
abstract class {
    public abstract void disp();
    int a;
    private int num(int n)
        Return n*10;
}
	19. Example:
	interface A{
	    public abstract void show();
	    public abstract int num();
	}